



**Jaime Jacinto das
Neves Marques**

Relatório de Atividade Profissional



**Jaime Jacinto das
Neves Marques**

Relatório de Atividade Profissional

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Ernesto Ventura Martins, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro. A este respeito, a legislação estabelece o seguinte:

Processos que potenciam a obtenção do grau de mestre Documento orientador

De acordo com o Decreto-Lei 74/2006, “o ciclo de estudos conducente ao grau de mestre integra nomeadamente uma dissertação de natureza científica ou um trabalho de projeto, originais e especialmente realizados para este fim, ou um estágio de natureza profissional objeto de relatório final, consoante os objetivos específicos visados, nos termos que sejam fixados pelas respetivas normas regulamentares.

A elaboração da dissertação ou do trabalho de projeto e a realização do estágio são orientadas por doutor ou por especialista de mérito reconhecido como tal pelo órgão científico estatutariamente competente do estabelecimento de ensino superior, nacional ou estrangeiro. A orientação pode ser assegurada em regime de co-orientação, quer por orientadores nacionais, quer por nacionais e estrangeiros.”

(...)

2. Adopção das recomendações do Conselho de Reitores das Universidades Portuguesas (CRUP) para licenciados possuidores de um ciclo de estudos pré-Bolonha que desejam obter o grau de mestre.

Na sequência da recomendação do CRUP de 8 de Janeiro de 2011, sobre o estímulo à obtenção do grau de mestre numa especialidade por parte de diplomados que tenham terminado as suas licenciaturas ao abrigo de graus anterior ao Processo de Bolonha, estabelece-se o seguinte:

- 2.1 Os diplomados que tenham terminado as suas licenciaturas ao abrigo do sistema de graus anterior ao Processo de Bolonha e que pretendem obter o grau de Mestre em área científica afim à da licenciatura, poderão inscrever-se nesse ciclo de estudos, solicitando a creditação da formação adquirida na licenciatura na componente letiva do respetivo mestrado, e realizando uma dissertação de pendor científico ou um projeto, nos termos da legislação em vigor.

- 2.2 Os diplomados que tenham terminado as suas licenciaturas ao abrigo do sistema de graus anterior ao Processo de Bolonha, e que pretendem obter o grau de Mestre em área científica afim à da licenciatura, e que, para além disto, tenham mais de 5 anos de experiência profissional relevante, poderão inscrever-se nesse ciclo de estudos, solicitando a creditação da formação adquirida na licenciatura na componente letiva do respetivo mestrado, e realizando uma dissertação de pendor científico, nos termos da legislação em vigor, ou poderão ainda optar, em alternativa às opções referidas no ponto anterior, pela apresentação de um relatório detalhado sobre a sua atividade profissional. Este relatório detalhado consistirá num trabalho que evidencie o percurso profissional do diplomado e as competências adquiridas no âmbito da especialidade do Mestrado em que o estudante se encontra inscrito demonstrando o domínio conceptual e metodológico próprio da respetiva área científica. A elaboração deste relatório será orientada por doutor ou por especialista de mérito reconhecido como tal pelo órgão científico estatutariamente competente do estabelecimento de ensino superior, nacional ou estrangeiro. Este relatório será objeto de provas científicas, que incluirá a discussão das experiências e competências adquiridas.

o júri

presidente

Prof. Doutor Rui Manuel Escadas Ramos Martins
Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e
Informática da Universidade de Aveiro

Professor Doutor Valter Filipe Miranda Castelão da Silva
Professor Adjunto da Escola Superior de Tecnologia e Gestão de Águeda

Professor Doutor Ernesto Fernando Ventura Martins
Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e
Informática da Universidade de Aveiro

Agradecimentos

- Dr.^a R. Marques – Psicóloga
- Orientador – Professor Doutor Ernesto Ventura Martins
- Universidade de Aveiro, mais particularmente ao Departamento de Eletrónica, Telecomunicações e Informática
- Dr. C. Anachoreta e Instituições relacionadas
- Eng. C. Marques

Minha Mãe

A todos,

MUITO OBRIGADO.

Jaime Jacinto das Neves Marques

palavras-chave

Comunicações óticas (controlo de atenuação, célula Peltier), gestão de redes (gateway, SNMP), rede telefónica fixa (teste, manutenção), gestão informática.

resumo

Este documento é um relatório da atividade profissional desenvolvida pelo autor no período Setembro de 1996 a Novembro de 2005.

O Capítulo 1 faz o enquadramento geral do documento e situa no tempo e no espaço cada uma das atividades reportadas nos capítulos seguintes.

O Capítulo 2 descreve a atividade do autor na função de hardware developer. Descreve o trabalho efetuado na implementação de um módulo que controla a potência relativa e absoluta de um conjunto de lasers em oito fibras óticas. O módulo tem diversas aplicações tais como correcção de não-linearidades do espectro ótico e adaptação de potências óticas.

O Capítulo 3 descreve a atividade do autor enquanto software developer para gestão de redes. Descreve um sistema de gestão de redes e o software construído para esse fim.

O Capítulo 4 descreve a atividade profissional do autor no desenvolvimento de software para a rede telefónica fixa. O software destinou-se a teste e manutenção de linhas telefónicas e telefones.

O Capítulo 5 descreve a atividade profissional do autor na informatização de laboratórios de análises clínicas. O autor implementou software para comunicações entre os equipamentos que fazem as análises e uma base de dados.

Finalmente, os Capítulos 6 e 7 descrevem atividades profissionais em que o autor esteve envolvido por períodos mais curtos.

keywords

Optical communications (attenuation control, Peltier Cells), network management (gateway, SNMP), PSTN and ISDN (test and maintenance), application software.

abstract

This document is a report that describes the work developed by the author, during the period from September 1996 to November 2005, as an Electronic and Telecommunication Engineer.

Chapter 1 gives a general framework of this report.

Chapter 2 describes an optical communications module that attenuates and controls the power of a set of lasers.

Chapter 3 describes the author's work as a software developer in network management applications.

Chapter 4 describes the author's work as a software developer in PSTN and ISDN environments.

Chapter 5 describes management software developed by the author for clinical analysis laboratories.

Finally, Chapters 6 and 7 describe other professional activities in which the author was involved for shorter periods of time.

My way or the highway

Provérbio inglês

Índice

Índice de Figuras.....	xvii
Índice de Tabelas.....	xviii
Acrónimos e Siglas	xix
Capítulo 1 Introdução.....	1
Capítulo 2 VOA Card	3
2.1. Local e período em que decorreu.....	3
2.2. Introdução	3
2.3. Objetivos e especificações.....	4
2.4. Responsabilidades e competências específicas.....	6
2.5. Enquadramento teórico	6
2.6. Projeto e implementação.....	7
2.6.1. O08VA-1 – Descrição sumária	7
2.6.1.1. O08VA-1 com o VOA DUPONT	8
2.6.1.2. O08VA-1 com o VOA GEMFIRE.....	11
2.6.2. Controlo da atenuação.....	13
2.6.2.1. Introdução	13
2.6.2.2. A precisão	14
2.6.2.3. A Resolução	17
2.6.2.4. Controlo da atenuação em O08VA-1 / DUPONT.....	17
2.6.2.5. Controlo da atenuação em O08VA-1 / GEMFIRE	20
2.6.3. Controlo da temperatura dos VOAs.....	23
2.6.3.1. Introdução	23
2.6.3.2. Controlo da temperatura do VOA DUPONT	26
2.6.3.3. Controlo da temperatura do VOA GEMFIRE	28
2.6.4. Calibração	30
2.6.4.1. Calibração do VOA DUPONT	30
2.6.4.2. Calibração do VOA GEMFIRE	32

2.7.	Testes e Resultados	32
2.8.	Bibliografia.....	34
Capítulo 3	Gestão de Redes.....	35
3.1.	Local e período em que decorreu	35
3.2.	Introdução	35
3.3.	Objetivos e especificações	36
3.4.	Responsabilidades e competências específicas	37
3.5.	Enquadramento teórico	37
3.5.1.	SNMP	38
3.5.1.1.	Mensagens e <i>Traps</i>	39
3.5.1.2.	Base de dados MIB	40
3.5.1.3.	Segurança	41
3.6.	Projeto e implementação	42
3.6.1.	Introdução	42
3.6.2.	As BOAMTasks.....	42
3.6.2.1.	BOAMTasks – estrutura	42
3.6.2.2.	BOAMTasks na <i>gestão da rede</i>	43
3.6.2.3.	<i>Tasks</i> para segurança das comunicações – <i>Tunneling</i>	46
3.6.2.4.	As BOAMTasks no Sistema	47
3.6.3.	Aplicações para gestão da rede	47
3.6.3.1.	O <i>NetManager</i>	47
3.6.3.2.	O <i>Containment View</i>	48
3.7.	Resultados	48
3.8.	Bibliografia.....	48
Capítulo 4	Software para a Rede Telefónica Fixa	51
4.1.	Local e período em que decorreu	51
4.2.	Introdução	51
4.3.	Objetivos e especificações	52
4.4.	Responsabilidades e competências específicas	52
4.5.	Enquadramento teórico	53
4.6.	Projeto e implementação	55
4.6.1.	O Equipamento	55

4.6.2.	O Software	57
4.6.2.1.	Subsistemas JGAMD e JTLMD	59
4.6.2.2.	A Customer Application.....	59
4.7.	Resultados	60
4.8.	Bibliografia	60
Capítulo 5 Informatização de Laboratórios de Análises Clínicas		61
5.1.	Local e período em que decorreu.....	61
5.2.	Introdução	61
5.3.	Objetivos e especificações	62
5.4.	Responsabilidades e competências específicas.....	63
5.5.	Enquadramento teórico	63
5.6.	Projeto e implementação.....	64
5.6.1.	Introdução	64
5.6.2.	O Equipamento	65
5.6.3.	O Software do Sistema Operativo	66
5.6.4.	A Base de Dados	67
5.6.5.	Software de Comunicações com os analisadores.....	67
5.6.5.1.	Protocolos proprietários	69
5.6.5.2.	Protocolo <i>ASTM</i>	69
5.6.5.3.	Funcionamento <i>Batch Mode</i>	69
5.6.5.4.	Funcionamento <i>Host Query Mode</i>	69
5.6.6.	Outros dados	70
5.7.	Resultados	70
5.8.	Bibliografia	71
Capítulo 6 Automatismos		73
6.1.	Local e período em que decorreu.....	73
6.2.	Introdução	73
6.3.	Objetivos e especificações	74
6.4.	Responsabilidades e competências específicas.....	74
6.5.	Enquadramento teórico	75
6.6.	Projeto e implementação.....	75
6.6.1.	Introdução	75

6.6.2. O sistema de desenfumagem elétrico	75
6.7. Alguns Clientes.....	77
6.8. Resultados	77
6.9. Bibliografia.....	77
 Capítulo 7 Docência no Ensino Secundário	79
 Apêndice A – MIB-II (excertos).....	81
 Apêndice B – Boletim Clínico	85

Índice de Figuras

Figura 2.1 – O08VA-1 – símbolo.....	4
Figura 2.2 – O08VA-1 – Painel Frontal.....	7
Figura 2.3 – O08VA-1 com VOA DUPONT – Diagrama de Blocos.....	9
Figura 2.4 – O08VA-1 com VOA GEMFIRE – Diagrama de Blocos	12
Figura 2.5 – Tensões de referência	14
Figura 2.6 – O08VA-1 com VOA <i>Dupont</i> – curvas típicas da atenuação provocada pelo <i>Heater XA</i>	18
Figura 2.7 – O08VA-1 com DUPONT – DACs de controlo da atenuação.....	19
Figura 2.8 – O08VA-1 com DUPONT – andar final de controlo da atenuação – canais 1 e 2	19
Figura 2.9 – VOA DUPONT – <i>Pinout</i> e ligações	20
Figura 2.10 – VOA GEMFIRE – modelo elétrico dos <i>heaters</i> dos canais	20
Figura 2.11 – VOA GEMFIRE – curvas típicas	21
Figura 2.12 – O08VA-1 com GEMFIRE – circuito de saída de controlo da atenuação.....	21
Figura 2.13 – O08VA-1 com GEMFIRE – circuito de controlo – DACs.....	22
Figura 2.14 – O08VA-1 com GEMFIRE – circuito de controlo – saídas para canais 0:3	22
Figura 2.15 – VOA GEMFIRE – <i>Pinout</i> e ligações.....	23
Figura 2.16 – Controlo da temperatura dos VOAs – princípio	24
Figura 2.17 – DACs e ADCs para controlo da temperatura dos VOAs DUPONT e GEMFIRE.....	25
Figura 2.18 – <i>Thermoelectric Cooler</i> (TEC)	26
Figura 2.19 – Circuito de controlo do TEC.....	27
Figura 2.20 – Controlo da temperatura do VOA GEMFIRE	29
Figura 2.21 – Configuração ótica de um agrupamento de <i>N</i> atenuadores.....	34
Figura 3.1 – <i>Gateway</i>	36
Figura 3.2 – O sistema SNMP	39
Figura 3.3 – <i>Gestão de Redes</i> – as equipas de desenvolvimento	42
Figura 3.4 – As BOAMTasks na gestão da Rede	43
Figura 3.5 – Túnel <i>ESP</i>	46
Figura 3.6 – Túnel L2TP/IPSec.....	46
Figura 3.7 – O Sistema final	47
Figura 4.1 – Telefone Tipo 7P – esquema parcial	54
Figura 4.2 – A Central Telefónica – diagrama de blocos	55
Figura 4.3 – Sistema de teste de linha e telefone	58
Figura 5.1 – O Sistema Informático de Análises Clínicas	65
Figura 5.2 – Sistema de análises clínicas – o equipamento	66
Figura 5.3 – Fluxograma do software de comunicações.....	68
Figura 6.1 – Sistema de desenfumagem elétrico.....	76
Figura 6.2 – Atuador instalado em janela	76

Índice de Tabelas

Tabela 2.1 – Requisitos Óticos.....	5
Tabela 2.2 – O08VA-1 – especificações	5
Tabela 2.3 – Especificações do VOA DUPONT	8
Tabela 2.4 – Especificações do VOA GEMFIRE	11
Tabela 2.5 – TL431A – Caraterísticas elétricas.....	15
Tabela 2.6 – Precisão da Atenuação.....	16

Acrónimos e Siglas

ADSL	<i>Asynchronous Digital Subscriber Line</i>
ASTM	<i>American Society for Testing and Materials</i>
CS	<i>Chip Select</i>
DSL	<i>Digital Subscriber Line</i>
ESP	<i>Encapsulating Security Payload</i>
FDDI	<i>Fiber Distributed Data Interface</i>
ICMP	<i>Internet Control Message Protocol</i>
ISDN	<i>Integrated Services Digital Network</i>
L2TP	<i>Layer 2 Tunneling Protocol</i>
LASER	<i>Light Amplification by Stimulated Emission of Radiation</i>
MIB	<i>Management Information Base</i>
NC	<i>Not Connected</i>
NTC	<i>Negative Temperature Coefficient</i>
OPAMP	<i>Operational Amplifier</i>
OSI	<i>Open Systems Interconnection</i>
POTS	<i>Point Of Termination Station</i>
PSTN	<i>Public Switched Telephone Network</i>
RDIS	<i>Rede Digital com Integração de Serviços</i>
RFC	<i>Request for Comment</i>
SNMP	<i>Simple Network Management Protocol</i>
SPI	<i>Serial Peripheral Interface</i>
TEC	<i>Thermoelectric Cooler</i>
UDP	<i>User Datagram Protocol</i>
VOA	<i>Variable Optical Attenuator</i>

Capítulo 1

Introdução

Este Documento insere-se no Curso de Mestrado Integrado em Engenharia Eletrónica e Telecomunicações, da Universidade de Aveiro. É o “relatório detalhado” sobre a atividade profissional do autor, desde que obteve, em 1995, a Licenciatura em Engenharia Eletrónica e Telecomunicações até ao presente.

O enquadramento legal está em “Documento Orientador”, emitido pela Universidade de Aveiro, cujo texto se reproduz parcialmente no início deste Documento.

Este Relatório está ordenado por capítulos. Cada capítulo descreve trabalhos que o autor realizou num local e período específicos. Está ordenado desde o mais recente para o mais antigo.

A última atividade profissional do autor, como licenciado, foi na empresa *Siemens SA*. Trabalhou nesta empresa de Abril de 1999 a Novembro de 2005.

Passou por três departamentos, enquanto esteve nesta empresa:

Nos anos de 2004 e 2005, trabalhou no departamento de Hardware, na função de *Hardware Developer*, onde executou vários trabalhos para Sistemas de Comunicação por Fibra Ótica. Este assunto está exposto no Capítulo 2.

Nos anos de 2002 e 2003, trabalhou noutro departamento onde desenvolvia e testava software para Gestão de Redes. Este assunto está exposto no Capítulo 3.

Desde Abril de 1999 até finais de 2001, ainda na empresa *Siemens*, desenvolveu e testou software para centrais telefónicas da rede fixa. Este assunto está exposto no Capítulo 4.

De Fevereiro de 1998 a Abril de 1999, trabalhou na empresa *Maxdata*, que informatiza Laboratórios de Análises Clínicas. A sua função na empresa foi a de implementar o software de comunicações entre um sistema de base de dados e os equipamentos que fazem as análises (os analisadores). Este assunto está exposto no Capítulo 5.

De finais de 1997 a Fevereiro de 1998, o autor trabalhou na empresa *DASEL – Desenvolvimento de Aplicações e Sistemas de Eletrónica, Lda.*, em Oeiras. A empresa instalava automatismos para desenfumagem, portas, grades, barreiras. O autor tinha como funções elaborar os projetos das instalações desses equipamentos, e coordenar a equipa que os instalava. Este assunto está exposto no Capítulo 6.

Como licenciado, a sua atividade profissional começou como Professor do Ensino Secundário, no Ano Letivo de 1996/1997, na Escola Secundária de Ourém. Leccionava, principalmente, a disciplina *Práticas Oficinais e Laboratoriais* ao 10º ano de Escolaridade. Nesta disciplina, ensinava, sobretudo, instalações elétricas e telefónicas.

No Ano Letivo de 1997/1998, foi “colocado” em Portalegre, a leccionar matemática no Ensino Básico. Esteve apenas alguns meses neste local. Em finais de 1997, saiu do ensino e foi trabalhar numa empresa. O assunto da sua atividade como docente está exposto no Capítulo 7.

A sua atividade como licenciado decorreu, portanto, no período de Setembro de 1996 a Novembro de 2005.

Mas, antes deste período, trabalhou na Assistência Técnica de equipamentos eletrónicos domésticos – TVs, Rádios, Leitores de CD, Videogravadores. Trabalhou na Assistência Técnica da *SONY* e da *SANYO*.

Também, desde 2007, até ao presente, por razões de força maior, faz alguns trabalhos de assistência técnica ao mesmo tipo de equipamentos, noutra situação.

Capítulo 2

VOA Card

2.1. Local e período em que decorreu

Este projeto decorreu na empresa *Siemens, SA*, nos anos de 2004 e 2005.

2.2. Introdução

Os Sistemas de Comunicação por Fibra Ótica são cada vez mais frequentes.

A fibra ótica começou por ser aplicada em comunicações de longa distância (superiores a 100Km), mas, atualmente, também já equipa redes metropolitanas.

A fibra ótica é um canal de comunicações, que se destaca pelas seguintes características:

- Elevada Largura de Banda;
- Imunidade a interferências eletromagnéticas;
- Elevada distância entre Repetidores (cerca de 100 km);

- Baixo custo.

No entanto, as fibras óticas, em longas distâncias, apresentam não-linearidades, sendo que alguns comprimentos de onda sofrem maior atenuação do que outros.

Por isso, é necessário compensar essas não-linearidades. A compensação pode ser realizada na origem – pré-ênfase – e/ou no destino – de-ênfase.

Por outro lado, quando se adiciona ou se retira tráfego aos sistemas de comunicação por fibra ótica, é necessário adaptar a potência do sinal ótico.

Neste projeto, desenvolveu-se um módulo, a que se chamou O08VA-1, para integrar um Sistema de Comunicações por Fibra Ótica. A O08VA-1 é uma *VOA (Variable Optical Attenuator) Card*, que tem o símbolo indicado na Figura 2.1.

A *Card* O08VA-1 possui 8 atenuadores óticos. Cada um dos 8 atenuadores atenua a potência entre 0 e 20 dB de um laser presente num canal de comunicação por fibra ótica, independentemente.

Os atenuadores executam funções de pré-ênfase e de-ênfase, e podem atuar sobre um canal ou sub-banda.

De uma forma geral, o Atenuador permite controlar a potência de um laser em fibra ótica.

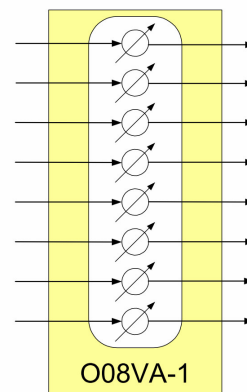


Figura 2.1 – O08VA-1 – símbolo

2.3. Objetivos e especificações

Analisadas as características do Sistema de Comunicações por Fibra Ótica, verificou-se a necessidade de ter um módulo (uma *Card*) que faça a atenuação de um conjunto de lasers em fibras óticas, de modo a poder controlar-se a potência absoluta e a potência relativa entre eles. Assim, definiram-se os *Requisitos* que um módulo a construir deve ter – ver Tabela 2.1.

Pretende-se um módulo, para ser instalado em *Rack*, que faça a atenuação controlada, independente, de 8 canais de comunicação por fibra ótica.

Item	Symbol	Conditions	Min	Typ	Max	Unit
Channel count	N	nominal	8			-
Signal band	v_C		192.1	-	196.0	THz
Insertion loss for signal path (connector to connector)	IL		-	-	2.3	dB
Dynamic Range	ΔA_{\max}		20	20	-	dB
Response time	τ	10% to 90% or 90% to 10%			10	ms
Temperature dependence of attenuation	TDL	relative to $T_{RT} \pm 3^\circ\text{C}$; within ΔA_{\max}	-	-	± 0.1	dB
Wavelength dependence of attenuation	WDL	within v_C - for min IL - for $0 \text{ dB} < A \leq 10 \text{ dB}$ - for $10 \text{ dB} < A \leq 20 \text{ dB}$ - for $20 \text{ dB} < A \leq \Delta A_{\max}$	- - - -	0.1 0.2 0.4 0.4	0.2 0.3 0.6 0.6	dB
Wavelength dependent ripple	WDR	measured loss difference in 0.4nm sliding window; within ΔA_{\max}	-	0.01	0.02	dB
Attenuation resolution	ΔA_{res}		-	-	0.05	dB
Absolute setting accuracy	ΔA_{set}		-	-	± 0.1	dB
Reproducibility	ΔA_{REP}		-	-	0.1	dB
Attenuation drift	ΔA_{drift}		-	-	± 0.5	dB
	ΔA_{time}	constant voltage and temperature - over 1 hour - over 1 day	- -	± 0.05 ± 0.05	± 0.1 ± 0.1	dB/h dB/d
Return loss	RL	excluding connector	40	45	-	dB
Optical crosstalk	XT_{opt}			-50	-45	dB
Thermal crosstalk	XT_{therm}			0.1	0.2	dB
Maximum input power	$P_{\text{max, 1ch}}$	single channel within v_C	-	-	19	dBm
	$P_{\text{max, 4ch}}$	4 adjacent channels within v_C	-	-	21	dBm

Tabela 2.1 – Requisitos Óticos

Analisados os *Requisitos*, e confrontados com o material disponível, fez-se a especificação do módulo – Tabela 2.2.

O08VA-1 – especificações	
Número de canais	8
Gama de atenuação	0 – 20 dB
Banda de funcionamento	1528 – 1610 nm
Perdas de inserção máximas	1.5 dB
Resolução	0.1 dB por <i>step</i>
Tempo de resposta	10 ms
Potência máxima de entrada por canal	<21 dBm

Tabela 2.2 – O08VA-1 – especificações

Note-se que a *Attenuation resolution* passou de 0.05 dB para 0.1 dB. À partida, já era incoerente ter uma *Attenuation resolution* de 0.05 dB, quando a *Absolute setting accuracy* era de ± 0.1 dB – ver Tabela 2.1.

Por isso e pelas características dos componentes que serão analisadas na Secção 2.6.2.2, e pelo exposto na Secção 2.6.2.3, especificou-se para o módulo uma *Attenuation resolution* de 0.1 dB.

2.4. Responsabilidades e competências específicas

Neste projeto o autor trabalhou no desenvolvimento do hardware da *O08VA-1*.

A partir dos objetivos e especificações (ver Secção 2.3), elaborou-se um circuito, uma solução, que satisfizesse.

Após a esquematização do circuito, no computador, este foi passado para a equipa que faz os *layouts*.

O *layout* da *O08VA-1* foi enviado para a Alemanha, para fabrico do hardware.

Disponibilizado o hardware, fizeram-se testes e correcções.

Verificadas as anomalias, foi feita nova produção do hardware, com as correcções necessárias.

O desenvolvimento terminou com os testes ao hardware final, e atualização da documentação.

2.5. Enquadramento teórico

A necessidade de controlar e adaptar a potência de um laser é frequente, por diversas razões, algumas das quais referidas nas secções anteriores.

O laser é um raio de luz que, através de métodos como a difração, é atenuado no seu percurso.

A difração num polímero termo-ótico pode ser controlada através do controlo da temperatura do polímero. Por sua vez, a temperatura do polímero pode ser controlada eletricamente.

É com base neste princípio que se desenvolveram dispositivos que permitem controlar a potência de um laser, eletricamente.

O módulo que se descreve neste capítulo permite controlar a potência de 8 sinais de luz laser.

2.6. Projeto e implementação

2.6.1. O08VA-1 – Descrição sumária

Fisicamente, o módulo O08VA-1 é uma placa a instalar em *Rack*, num Sistema de Comunicações por Fibra Ótica. Esta contém, por isso, um *Backplane Connector* com ligações elétricas para alimentação e controlo, e um Painei Frontal – ver Figura 2.2 – que contém as ligações óticas para *input* dos lasers a serem atenuados e *ouput* dos mesmos já atenuados.

O componente mais específico da O08VA-1 é o VOA (*Variable Optical Attenuator*), sendo este o elemento que determina fundamentalmente o seu desempenho. Os VOAs são dispositivos termo-óticos que atenuam a luz laser através do aquecimento de um canal ótico, especialmente desenvolvido para este fim. Controlando eletricamente o aquecimento deste canal, controla-se a atenuação do laser entre 0 e 20 dB.

A empresa encontrou dois fabricantes – *DUPONT* e *GEMFIRE* – que disponibilizam VOAs adequados aos objetivos

Os VOAs da DUPONT e da GEMFIRE não são, porém, compatíveis em termos elétricos.

Após estudos que incidiram sobretudo em fatores económicos, resolveu-se optar por uma única Placa, capaz de suportar ambos os módulos de VOA, em vez de construir duas placas distintas. Assim, a mesma placa – mesmo hardware e mesmo software – suporta o VOA da DUPONT ou o VOA da GEMFIRE. O software da O08VA-1 identifica qual o VOA instalado.

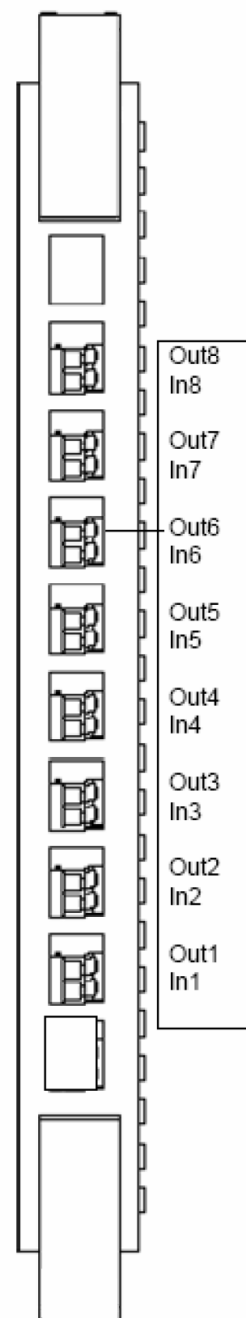


Figura 2.2 – O08VA-1 – Painei Frontal

2.6.1.1. O08VA-1 com o VOA DUPONT

A Tabela 2.3 elenca as características do VOA DUPONT, especificadas pelo fabricante. Como se observa, satisfazem os objetivos apresentados na Tabela 2.2.

Especificações do VOA DUPONT	
Número de canais	8
Gama de atenuação	0 – 20 dB
Banda de funcionamento	1528 – 1610 nm
Perdas de inserção	1.2 dB
Resolução	0.1 dB por <i>step</i>
Tempo de resposta	3 ms
Consumo @20dB	40 mW / canal
Atenuação quando sem alimentação	≈ 7 dB

Tabela 2.3 – Especificações do VOA DUPONT

A Figura 2.3 representa o Diagrama de Blocos da O08VA-1 com o VOA DUPONT.

O *Backplane Connector* faz a ligação elétrica da O08VA-1 ao *Rack*. Contém as ligações para alimentação e controlo da O08VA-1.

O *Card Controller* controla a O08VA-1. É constituído pelo CPU e pelas memórias. O *Card Controller* recebe, do *Backplane*, informação sobre a atenuação a aplicar a cada um dos 8 canais óticos, e configura a DAC respetiva, para que forneça a tensão que provoque a atenuação pretendida.

O Bloco *Power Supply* alimenta a O08VA-1. É constituído por um conjunto de conversores DC/DC que, a partir de 48 Volt vindos do *Backplane*, gera as tensões de alimentação necessárias: 5, 3.3, e 1.8V.

As tensões 3.3V e 1.8V alimentam o *Card Controller*. As tensões 5V e 3.3V alimentam os periféricos: VOA, DACs, ADCs.

Este Bloco tem associado a si 4 ADCs através das quais o *Card Controller* monitoriza as tensões da fonte. Se alguma estiver fora dos limites predefinidos, é feito *reset* da O08VA-1 e gerado um alarme.



O *Ethernet Transceiver* liga o *Card Controller* ao *Backplane*. Disponibiliza uma ligação *Ethernet 100baseT* para comunicações entre o *Card Controller* e o sistema. Permite efetuar a configuração da O08VA-1, e outras operações, tal como mudar a atenuação dos canais.

O VOA é o Bloco que faz a atenuação dos lasers.

O VOA DUPONT divide cada canal ótico em dois ramos, a que chama *exhaust arm* e *output arm*. A cada um dos ramos está associado um *heater* que os faz aquecer. Ao *output arm* está associado o *through heater* que, quando controlado/alimentado com tensões de 0 a 3.1V, provoca atenuação do sinal ótico de 7dB até à atenuação máxima (≈ 20 dB). Ao *exhaust arm* está associado o *exhaust heater* que, quando controlado/alimentado com tensões de 0 a 3.1V, provoca atenuação do sinal ótico de 7 a 0dB.

Sem qualquer controlo/alimentação, a atenuação é, aproximadamente, 7 dB.

As DAC 0-7 são usadas para o controlo da atenuação. O CPU envia às DACs informação sobre a atenuação a aplicar aos respetivos canais, para que as DACs forneçam tensões aos *heaters* associados a cada canal. Um Bloco DAC 0-7 controla os *Through heaters* e o outro Bloco DAC 0-7 controla os *Exhaust heaters*.

O *TEC Controller* serve para controlar a temperatura global do VOA, aquecendo e arrefecendo. Para se obter a precisão adequada na atenuação, é necessário que o VOA esteja a uma temperatura controlada. Através de uma DAC o *Card Controller* estabelece a temperatura a que o VOA deve funcionar.

Através da ADC o *Card Controller* acede à informação, provinda (ou não) do termistor do VOA. Esta informação é usada para dois fins:

- Detetar excessos de temperatura do VOA. Em caso de excesso de temperatura, é enviado ao sistema, através da interface *Ethernet*, um alarme.
- Na inicialização da O08VA-1, esta informação serve para identificar qual o VOA instalado (se o DUPONT, se o GEMFIRE).

2.6.1.2. O08VA-1 com o VOA GEMFIRE

A Tabela 2.4 elenca as características técnicas do VOA GEMFIRE, indicadas pelo fabricante. Como se pode constatar estas especificações satisfazem os objetivos apresentados na Tabela 2.2.

Especificações do VOA GEMFIRE	
Número de canais	8
Gama de atenuação	0 – 20 dB
Banda de funcionamento	1510 – 1610 nm
Perdas de inserção	1.3 dB
Resolução	Contínua
Tempo de resposta	5 ms
Consumo @20dB	20 mW / canal
Atenuação quando sem alimentação	0 dB

Tabela 2.4 – Especificações do VOA GEMFIRE

A O08VA-1 com o VOA GEMFIRE é semelhante à O08VA-1 com VOA DUPONT – ver Secção 2.6.1.1. Existem, no entanto, diferenças ao nível do controlo do VOA.

A Figura 2.4 representa o Diagrama de Blocos da O08VA-1 com o VOA GEMFIRE. Aqui o Bloco *TEC Controller*, presente na O08VA-1 com o VOA DUPONT, foi substituído pelo *Substrate Heater Controller*. A função é a mesma, mas o *Substrate Heater Controller* apenas permite aumentar a temperatura do VOA.

O VOA GEMFIRE é um dispositivo constituído por um polímero termo-ótico, que atenua a luz por difração induzida via um gradiente de temperatura. A temperatura é controlada através de *heaters* associados a cada canal. Fazendo passar uma corrente de 0 a 10 mA por um *heater*, faz-se aumentar a temperatura, e, consequentemente, a atenuação do sinal ótico de 0 a 20 dB.

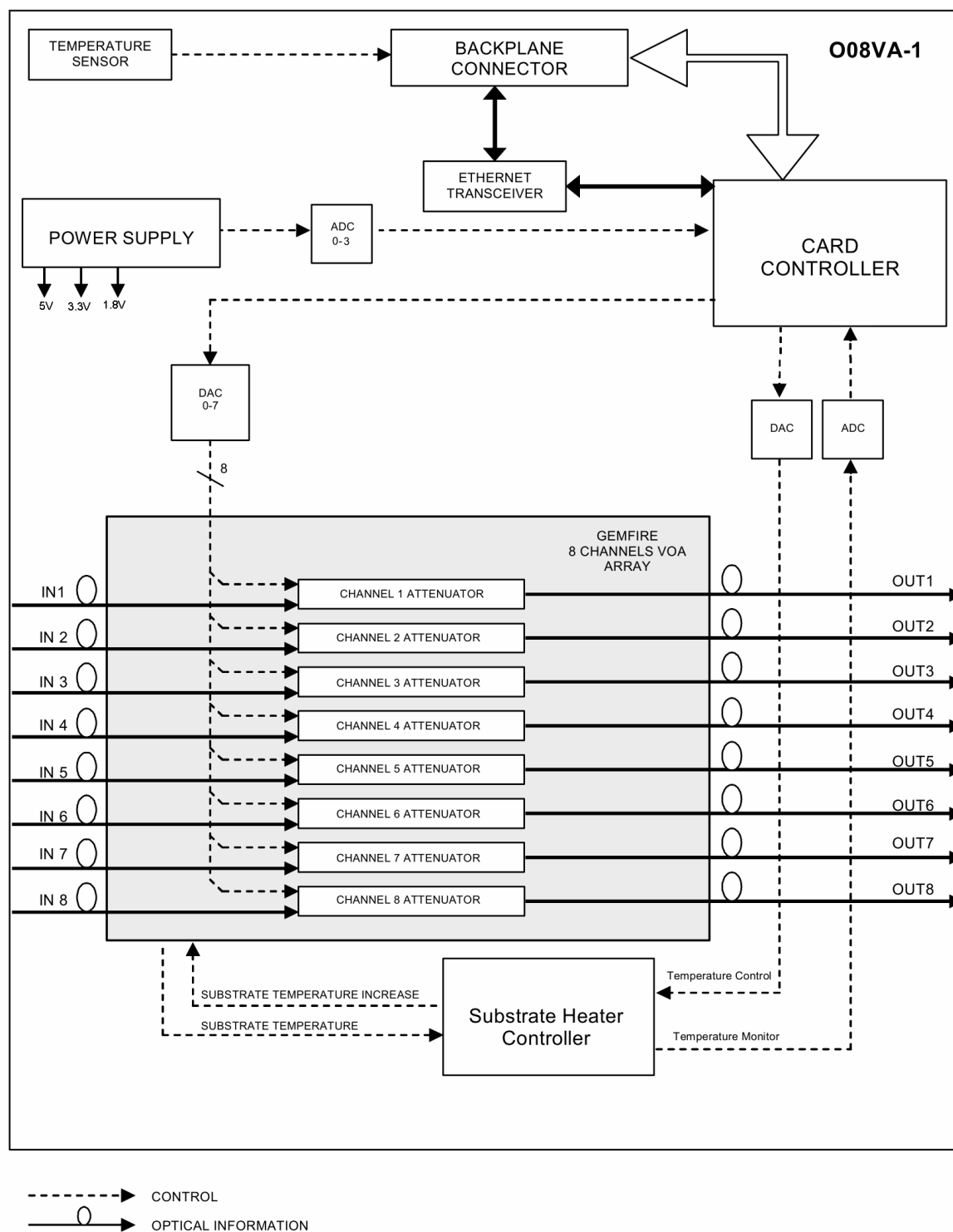


Figura 2.4 – O08VA-1 com VOA GEMFIRE – Diagrama de Blocos

2.6.2. Controlo da atenuação

2.6.2.1. Introdução

O controlo da atenuação é feito pelo sistema de comunicações onde o módulo for inserido. O controlo é efetuado por software, usando o Protocolo de Comunicações SNMP (*Simple Network Management Protocol*), através da interface *Ethernet* via *Backplane*.

Num Sistema SNMP há um *Agent* e um ou mais *Management Systems*.

O *SNMP Management System* envia ao *Agent* pedidos do tipo *Set*, *Get*, *GetNext*, *GetBulk*, de uma *variável* existente numa Base de Dados do *SNMP Agent*, chamada *MIB* (*Management Information Base*). O *Agent*, em resposta, executa as funções correspondentes, atualiza a *variável* na *MIB*, e envia uma Mensagem – Resposta – ao *SNMP Management System*.

No caso presente, o *SNMP Agent* é uma aplicação de software que está na O08VA-1, e o *SNMP Management System* é o Sistema de Comunicações onde está inserida a O08VA-1.

No Capítulo 3, o Protocolo SNMP é descrito mais em detalhe.

Para alterar a atenuação de um canal [0..7], o *SNMP Management System* envia ao *Agent* um Pedido *Set channelAttenuation* para um valor de atenuação na gama 0 a 20 dB.

O *Agent*, ao receber o Pedido, consulta uma *lookup table* na memória EEPROM da O08VA-1 para saber qual o valor a enviar à DAC respetiva, e envia esse valor à DAC. Em seguida, atualiza a *MIB* e envia resposta ao *SNMP Management System*.

A *lookup table* é um ficheiro de dados que o fabricante do VOA fornece juntamente com o VOA, e que contém os valores da atenuação em função do controlo (tensão se DUPONT, corrente se GEMFIRE).

O *SNMP Management System* também pode enviar ao *Agent* um Pedido *Get channelAttenuation*, para saber qual a atenuação que está a ser aplicada a um determinado canal.

Ao nível do hardware, o valor da atenuação aparece na DAC, que está associada ao canal.

Pretende-se uma Atenuação de 0 a 20 dB, com resolução de 0.1dB – ver Tabela 2.2. Em consequência, as DACs, através das quais é controlada a atenuação, devem ser, no mínimo, de 8 bits (256 níveis).

2.6.2.2. A precisão

Os VOAs

Face à elevada precisão e estabilidade dos VOAs, optou-se por operá-los em *loop* aberto, prescindindo-se de medidores de potência ótica.

No entanto, os fabricantes especificam degradação da precisão da atenuação em determinadas situações. A este respeito, a DUPONT afirma: «Se a potência total aplicada a um VOA for de 15 dBm a 21 dBm, e a atenuação for entre 10 e 20 dB, a imprecisão da atenuação sobe para $\pm 0.5\text{dB}$ ».

Tensões de referência

As tolerâncias das tensões de referência são outro factor de imprecisão. As DACs exigem tensões de referência de elevada precisão. As tensões mais críticas são as de referência para as DACs que fornecem as tensões de atenuação. Qualquer imprecisão nas tensões de referência vai-se refletir na imprecisão da atenuação ótica.

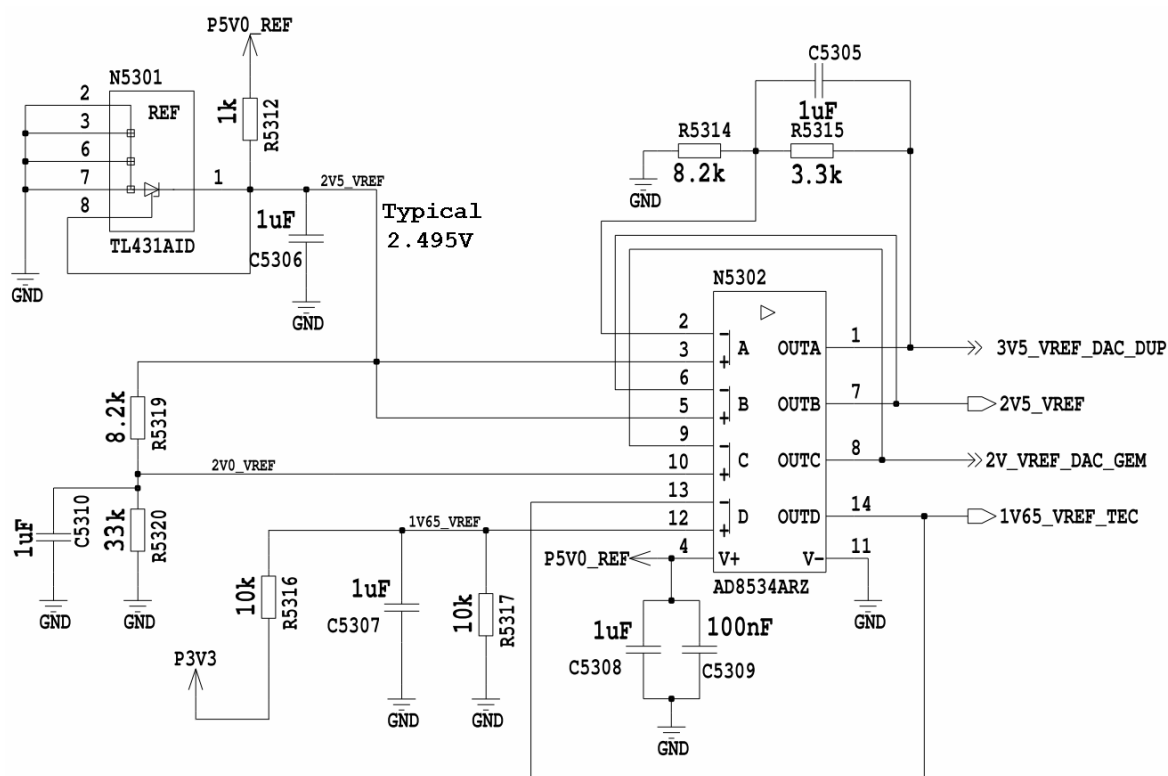


Figura 2.5 – Tensões de referência

As referências são obtidas a partir do *Programmable Shunt Regulator* TL431A – ver Figura 2.5.

Na O08VA-1 com o VOA DUPONT a referência é 3V5_VREF_DAC_DUP, que se pretende que seja de 3.5V – ver Figura 2.7. Na O08VA-1 com o VOA GEMFIRE a referência é 2V_VREF_DAC_GEM, que se pretende que seja de 2V – ver Figura 2.13.

A Tabela 2.5 indica as caraterísticas elétricas do componente TL431A, segundo o fabricante *FAIRCHILD SEMICONDUCTOR*. A Tabela foi extraída das *datasheets* do fabricante. Verifica-se que há todo um conjunto de parâmetros que afetam o valor da tensão de referência.

Parameter	Symbol	Conditions		TL431A			Unit
				Min.	Typ.	Max.	
Reference Input Voltage	VREF	VKA=VREF, IKA=10mA		2.470	2.495	2.520	V
Deviation of Reference Input Voltage Over-Temperature	$\Delta V_{REF}/\Delta T$	VKA=VREF, IKA=10mA TMIN≤TA≤TMAX		-	4.5	17	mV
Ratio of Change in Reference Input Voltage to the Change in Cathode Voltage	$\Delta V_{REF}/\Delta V_{KA}$	IKA=10mA	$\Delta V_{KA}=10V-V_{REF}$	-	-1.0	-2.7	mV/V
			$\Delta V_{KA}=36V-10V$	-	-0.5	-2.0	
Reference Input Current	IREF	IKA=10mA, R1=10KΩ,R2=∞		-	1.5	4	μA
Deviation of Reference Input Current Over Full Temperature Range	$\Delta I_{REF}/\Delta T$	IKA=10mA, R1=10KΩ,R2=∞ TA =Full Range		-	0.4	1.2	μA
Minimum Cathode Current for Regulation	IKA(MIN)	VKA=VREF		-	0.45	1.0	mA
Off - Stage Cathode Current	IKA(OFF)	VKA=36V, VREF=0		-	0.05	1.0	μA
Dynamic Impedance	ZKA	VKA=VREF, IKA=1 to 100mA f ≥1.0KHz		-	0.15	0.5	Ω

- TMIN= -25 °C, TMAX= +85 °C

Tabela 2.5 – TL431A – Caraterísticas elétricas

Para a obtenção das várias referências deve considerar-se como ponto de partida o ponto médio do intervalo entre VREF mínimo e máximo, que é 2.495V. As resistências do circuito que gera as tensões de referência, à excepção de R5312, são de precisão (tolerância de 0.1%).

Considerando como referência 2.495V, tem-se um desvio máximo de ±0.025V, que corresponde a ≈1% de 2.495V. Aliás, o fabricante do TL431A especifica uma *Output Voltage Tolerance* de 1%, para temperaturas entre -25 e +85°C.

Na situação extrema – de a atenuação ser de 20 dB – este desvio corresponde a uma imprecisão de 0.2 dB. Considera-se *situação extrema*, o caso de se tratar de módulo com VOA GEMFIRE. A imprecisão, na gama 0 a 2V, reflete-se na gama de atenuação 0 a 20 dB. No caso de se tratar de

um módulo com VOA DUPONT, o impacto da imprecisão da tensão de referência não é tão dramático, pois a atenuação está dividida em duas sub gamas.

Se a atenuação for de 5 dB, um desvio de 1% da tensão de referência corresponde a uma imprecisão de ± 0.05 dB.

As tolerâncias dos componentes, que constituem os circuitos que geram as tensões de referência para as DACs, provocam uma imprecisão que é proporcional às tolerâncias.

Os parâmetros da Tabela 2.5 afetam também a *Reproducibility* e a *Attenuation drift* dos VOAs – Tabela 2.1. Considerou-se, no entanto, como indicador genérico, a *Output Voltage Tolerance* especificada pelo fabricante do TL431A, na Secção *Ordering Information*, das *datasheets* deste componente, que é 1%.

Além da imprecisão do TL431A, há ainda a considerar as tolerâncias de outros componentes, tais como resistências, DACs.

Claramente, a precisão de 0.1 dB, de início requerida (ver Tabela 2.1), é irrealista nestas condições. Em primeiro lugar porque os VOAs não o possibilitam como vimos antes. Em segundo lugar, porque seria necessário outro circuito (mais dispendioso) para obter essa precisão - seria necessário implementar um *loop* ótico fechado.

Apresentados estes dados ao departamento da empresa que emitiu as especificações e os objetivos, fez-se uma revisão dos objetivos, no que respeita à precisão. Consideraram-se aceitáveis os valores que constam da Tabela 2.6, e que passaram a ser as especificações ao nível da precisão.

O mais frequente é necessitarem-se de atenuações até 10 dB, pelo que a imprecisão acrescida para atenuações acima de 10 dB, não se afigura problemática.

Attenuation (dB)	Absolute Accuracy (dB)
0-10	0.2
10-20	0.3 ¹⁾

Tabela 2.6 – Precisão da Atenuação

¹⁾ Para potências óticas de entrada superiores a 15 dBm, a imprecisão sobe para 0.7 dB

2.6.2.3. A Resolução

Por norma, se, por exemplo, a atenuação pretendida for de 6dB, e o *step* for 0.1dB, será de esperar que a atenuação obtida esteja entre 5.95 e 6.05 dB. O *step* é, no mínimo, o dobro do valor absoluto da precisão.

No caso presente, se se seguisse esta norma, ter-se-ia um *step* de $2 \times 0.3 = 0.6\text{dB}$, ou, se se quisesse contemplar todas as situações, ter-se-ia um *step* de $2 \times 0.7 = 1.4\text{dB}$ – ver Tabela 2.6.

Ora, se se pretender uma atenuação o mais próxima possível de, por exemplo, 6.3dB, e se o *step* fosse de 0.6dB, a *Attenuation setting* seria 6.6dB ou 6.0 dB; se fosse 6.6dB, resultaria numa atenuação que estaria no intervalo entre 6.4 a 6.8dB, considerando que a precisão, na gama de atenuações 0-10dB, é 0.2dB.

Mas, se o *step* for de 0.1dB, ainda que a precisão continue a ser 0.2dB, a *Attenuation setting* é de 6.3dB, resultando numa atenuação que está no intervalo 6.1 a 6.5dB; portanto, mais conveniente do que se o *step* fosse 0.6dB.

Considerando que o módulo O08VA-1 se destina a ser integrado num sistema de comunicações fabricado pela empresa, e de forma a aproveitar, ao máximo, as características da O08VA-1 e dos seus componentes, resolveu-se estabelecer uma *Resolução* de 0.1dB. De outra forma (com *step* elevado), haveria um desperdício de meios.

Ao efetuarem-se os testes à O08VA-1, deve-se ter sempre em conta a precisão – Tabela 2.6.

2.6.2.4. Controlo da atenuação em O08VA-1 / DUPONT

A atenuação de cada um dos 8 canais do VOA DUPONT é feita com um par de *heaters* (duas resistências): *Heater XA* (*Through heater*) e *heater XB* (*Exhaust heater*) – ver Figura 2.3.

Cada resistência dos *heaters* tem entre 60 e 120Ω.

A Figura 2.6 representa a curvas típicas da atenuação do *Heater XA*.

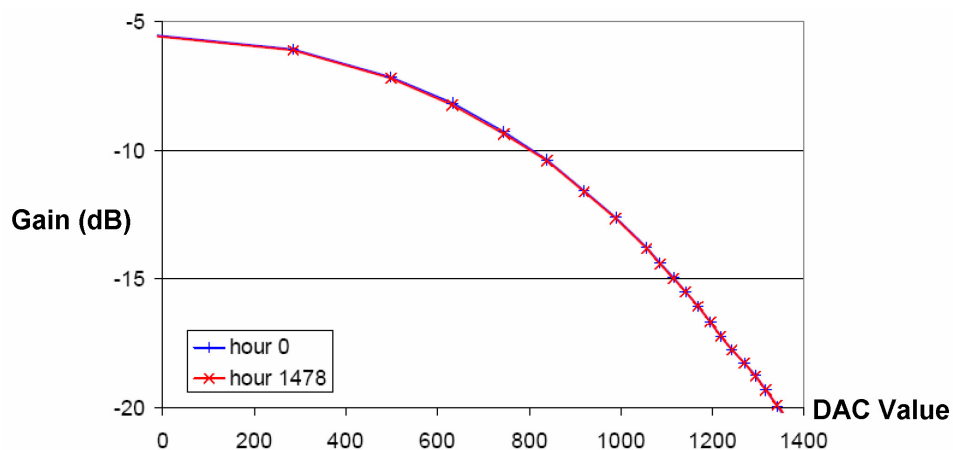


Figura 2.6 – O08VA-1 com VOA *Dupont* – curvas típicas da atenuação provocada pelo *Heater XA*.

Apenas um dos *heaters* é usado de cada vez. Quando nenhum dos *heaters* é alimentado, a atenuação é de ≈ 7 dB. O *Heater XB* faz diminuir a atenuação, e o *Heater XA* faz aumentar a atenuação.

Cada *heater* é controlado com uma tensão, que vai de 0 a 3.1V.

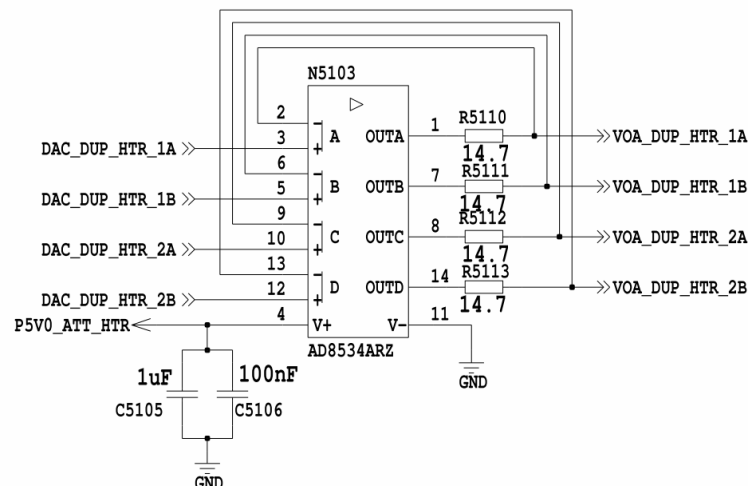
Se o *Heater XB* for alimentado/controlado com uma tensão de 0 a 3.1V, a atenuação produzida vai de 7 a 0 dB. Se o *Heater XA* for alimentado/controlado com uma tensão de 0 a 3.1V, a atenuação produzida vai de 7 até 20 dB.

A Figura 2.7 representa a conversão digital/analógico dos sinais de controlo da atenuação do VOA. Existem 16 DACs – 8 para controlar os *Heaters XA* e 8 para controlar os *Heaters XB*. O circuito integrado U5101 (MAX5306) contém um conjunto de 8 DAC's, para controlar a atenuação de 4 canais do VOA. U5102 controla a atenuação dos restantes 4 canais.

As DACs são controladas pelo CPU, através do barramento SPI.

Às DACs, é aplicada uma tensão de referência de 3.5V, suficiente para gerar a tensão de controlo de 0 a 3.1V, na saída.

Um simples *voltage follower*, com um OPAMP, tal como sugerido pelo fabricante do VOA, é suficiente para andar final de controlo do *heater* – ver Figura 2.8. Esta figura representa a implementação para os canais 1 e 2; para os restantes 3 a 8, a implementação é idêntica.



19

N5101, N5102, N5103 – ver Figura 2.8 – e N5104 contêm 4 OPAMPs cada, configurados como *voltage followers*, cujas entradas estão ligadas às saídas das DACs, cujas saídas são aplicadas aos *heaters* do VOA.

A Figura 2.9 representa o *Pinout* do VOA DUPONT, com indicação das ligações à *board* da O08VA-1.

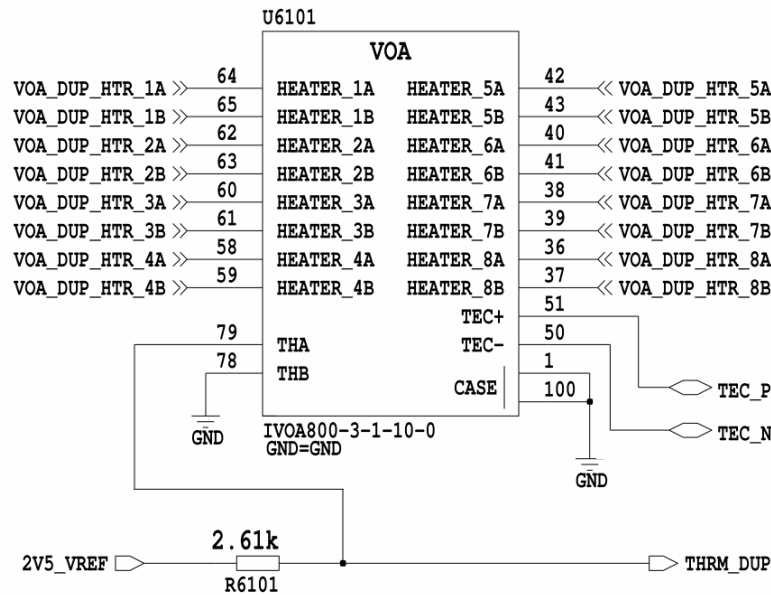


Figura 2.9 – VOA DUPONT – *Pinout* e ligações

2.6.2.5. Controlo da atenuação em O08VA-1 / GEMFIRE

A atenuação no VOA GEMFIRE é também controlada com *heaters*, um por canal.

A Figura 2.10 representa o modelo elétrico dos *heaters* dos canais, do VOA GEMFIRE.

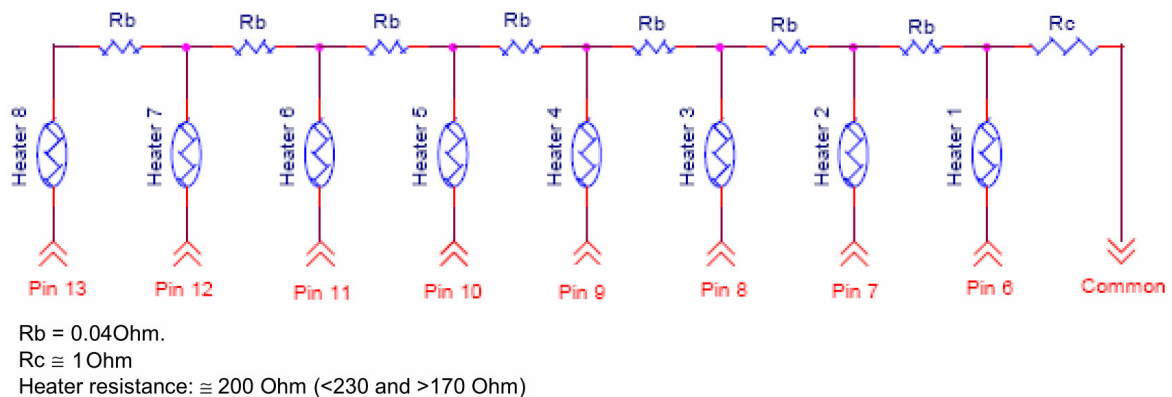


Figura 2.10 – VOA GEMFIRE – modelo elétrico dos *heaters* dos canais

Devido à existência das resistências R_b e R_c , não é adequado controlar os *heaters* com tensões, porque seria necessário reajustar a tensão de todos os canais, sempre que a atenuação de um deles fosse alterada. Por este motivo usou-se uma fonte de corrente por canal para o controle da atenuação.

A Figura 2.11 representa curvas típicas do Ganho/atenuação em função da corrente nos *heaters*. Com correntes de 0 a 10 mA obtêm-se atenuações de 0 a 20 dB.

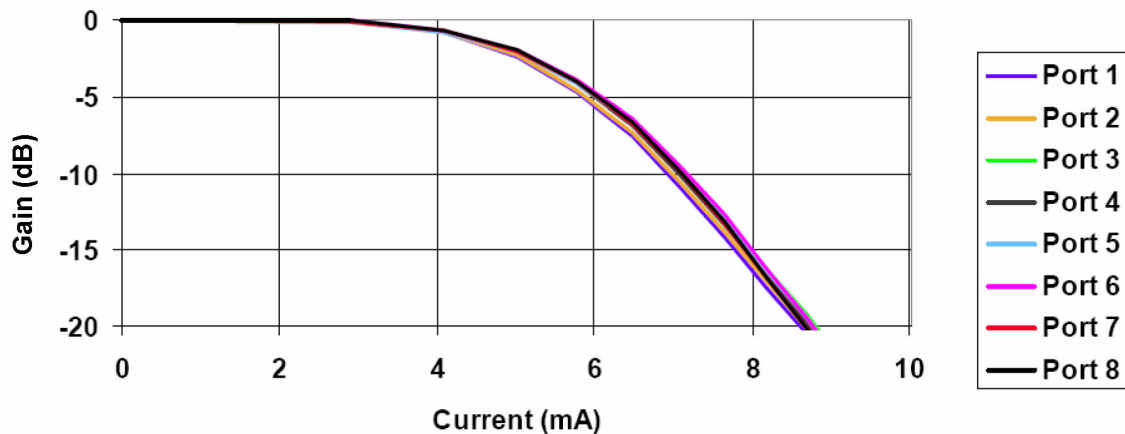


Figura 2.11 – VOA GEMFIRE – curvas típicas

A Figura 2.12 representa o circuito de controlo usado com cada *heater* (para cada canal). Com tensões de entrada de 0 a 2V, obtêm-se correntes de saída proporcionais, de 0 a 10 mA.

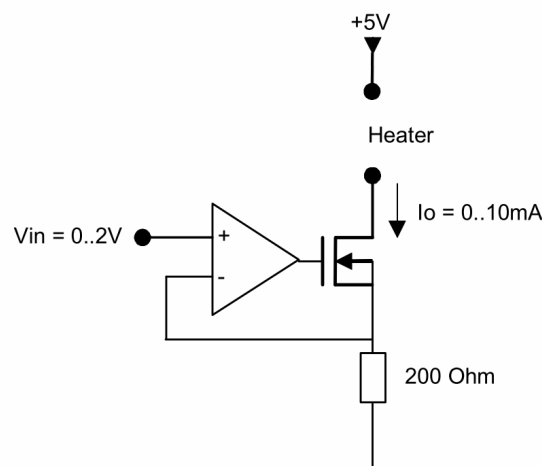


Figura 2.12 – O08VA-1 com GEMFIRE – circuito de saída de controlo da atenuação

A Figura 2.13 representa as DACs. Estas são controladas pelo CPU através do barramento SPI e possuem uma referência de 2V, que faz com que as saídas tenham uma excursão de sinal de 0 a 2V.

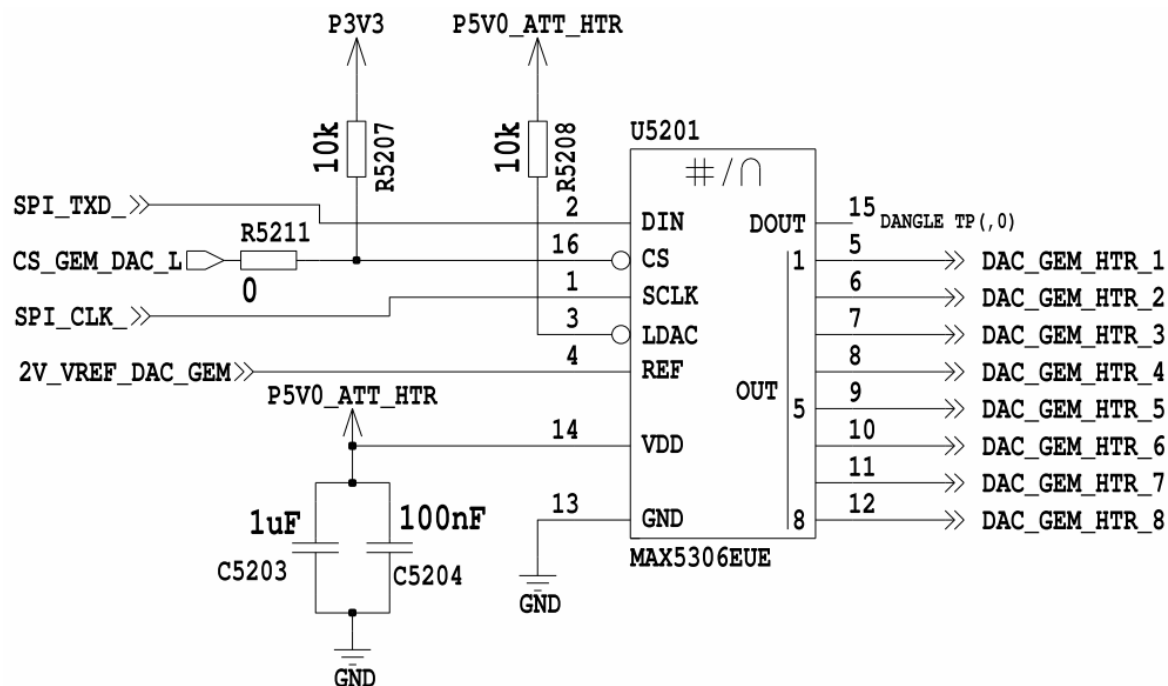


Figura 2.13 – O08VA-1 com GEMFIRE – circuito de controlo – DACs

A Figura 2.14 mostra a implementação final do circuito atrás descrito. Esta figura representa a implementação para os canais 0 a 3; para os restantes 4 a 7, a implementação é idêntica.

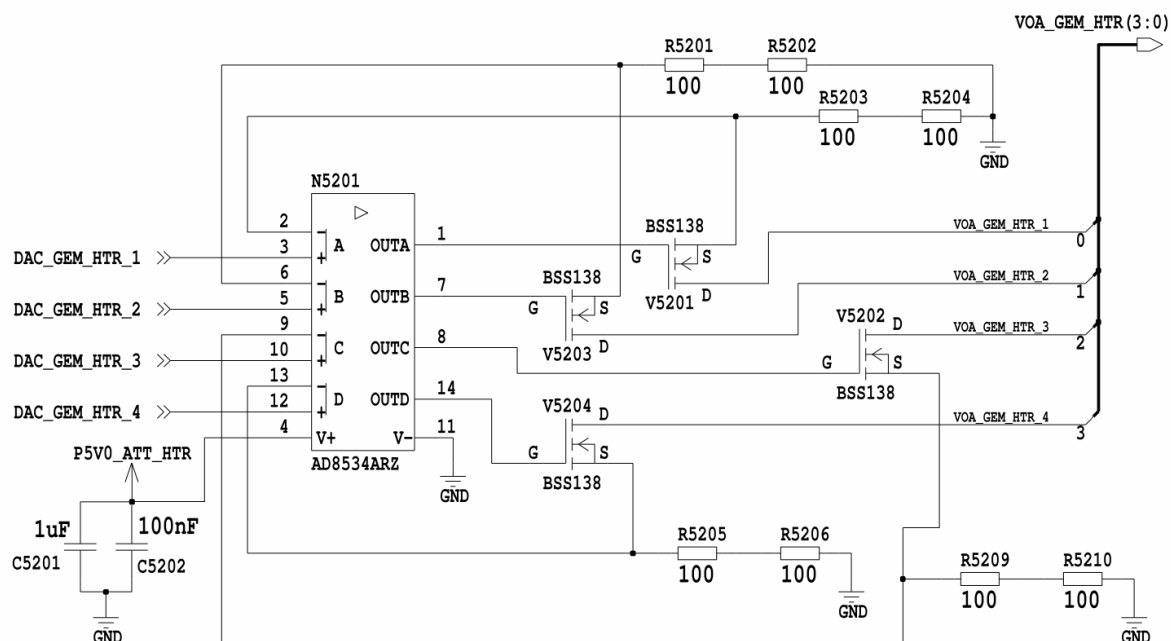


Figura 2.14 – O08VA-1 com GEMFIRE – circuito de controlo – saídas para canais 0:3

A Figura 2.15 representa o *Pinout* do VOA GEMFIRE, com indicação das ligações à *board* da O08VA-1.

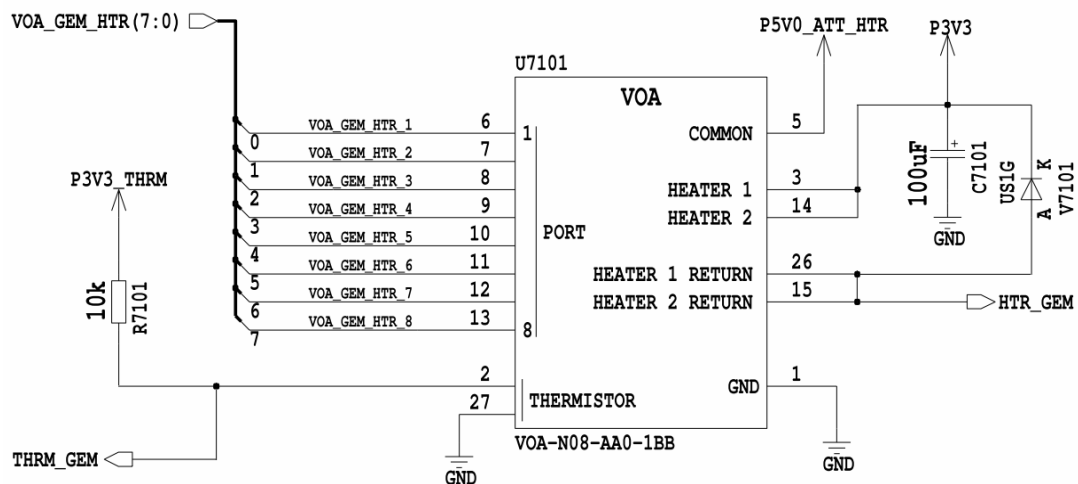


Figura 2.15 – VOA GEMFIRE – Pinout e ligações

2.6.3. Controlo da temperatura dos VOAs

2.6.3.1. Introdução

Para controlar a atenuação dos VOAs, tanto o da DUPONT como o da GEMFIRE, não basta controlar a temperatura de cada canal. É necessário controlar a temperatura global dos VOAs. Para se obter uma boa precisão na atenuação, é indispensável que a temperatura global dos VOAs se mantenha estável no valor fixado pelo fabricante. O controlo da temperatura global dos VOAs minimiza a *Thermal crosstalk*, a *Temperature dependence of attenuation*, a *Attenuation drift*, e permite uma *Reproducibility* precisa.

Os VOAs incluem dispositivos para alterar a sua temperatura e sensores para efetuar o seu controlo.

A temperatura ideal de funcionamento do VOA é programada pelo CPU. Uma DAC ligada ao CPU através do barramento SPI converte o valor binário (oriundo do CPU) numa tensão analógica que estabelece a temperatura de funcionamento do VOA – ver Figura 2.3 e Figura 2.4.

A monitorização da temperatura é feita através de um termistor e de uma ADC ligada ao CPU via SPI.

A ADC não fecha um *loop*. De facto, o controlo da temperatura do VOA não passa pelo CPU. A ADC serve, em primeiro lugar, para identificar qual o VOA instalado na O08VA-1 – se o modelo da DUPONT, se o da GEMFIRE. Na inicialização da O08VA-1, o software deteta a existência do

termistor, e, conseqüentemente, identifica qual o VOA instalado. A segunda função da ADC é informar o CPU da temperatura do VOA, para que, em caso anormal, seja ativado um alarme.

O controlo da temperatura do VOA, através do CPU, não é adequado. O CPU poderia bloquear, poderia não efetuar o controlo da temperatura atempadamente, e o VOA aquecer excessivamente. Tanto na O08VA-1 com o VOA DUPONT como na O08VA-1 com o VOA GEMFIRE, foi implementado um circuito local que faz o controlo. O CPU apenas indica a temperatura a que o VOA deve funcionar.

O sistema de controlo da temperatura interna do VOA é do tipo representado na Figura 2.16. O feedback negativo faz a temperatura do VOA ser a pretendida.

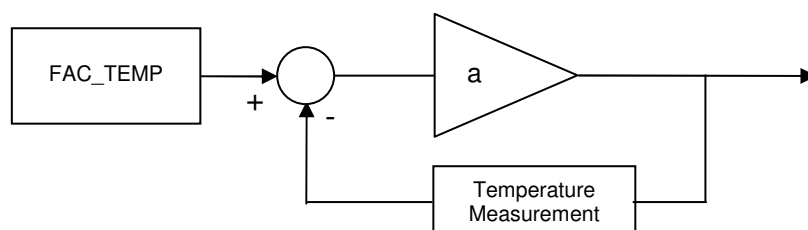


Figura 2.16 – Controlo da temperatura dos VOAs – princípio

O sinal “FAC_TEMP” (*Factory Temperature*) é o sinal analógico proveniente do CPU, através da DAC. Indica a temperatura a que o VOA deve funcionar e é um valor indicado pelo fabricante. A função *Temperature Measurement* é efetuada através de um termistor, incluído no VOA.

A Figura 2.17 representa o esquema final das interfaces entre o CPU e o *loop* de controlo da temperatura do VOA.

O componente central é o circuito integrado ADT7516ARQ, que possui 4 DACs de 12 bits e 4 ADCs de 10 bits. Duas DACs são usadas para o CPU fixar o valor da temperatura a que deve estar o VOA. Apenas uma destas DACs está ativa em cada O08VA-1.

Em relação às DACs, existem os seguintes sinais (ver também Figura 2.17), para o controlo da temperatura.

Para a O08VA-1 com o VOA DUPONT:

- PCU_TEC_CTR – é saída de DAC, que informa a temperatura a que deve estar o VOA DUPONT.
- 3V5_VREF_DAC_DUP – é tensão de referência de 3.5V; metade deste valor é referência para a DAC cuja saída é PCU_TEC_CTR. Com esta referência, a DAC tem uma saída PCU_TEC_CTR que vai de 0 a 1.75V.

Para a O08VA-1 com o VOA GEMFIRE:

- HTR_GEM_CTR – é saída de DAC, que informa a temperatura a que deve estar o VOA GEMFIRE.
- 2V5_VREF – é uma tensão de referência de 2.5V, para a DAC cuja saída é HTR_GEM_CTR. Com esta referência, a DAC tem um *output swing* HTR_GEM_CTR que vai de 0 a 2.5V

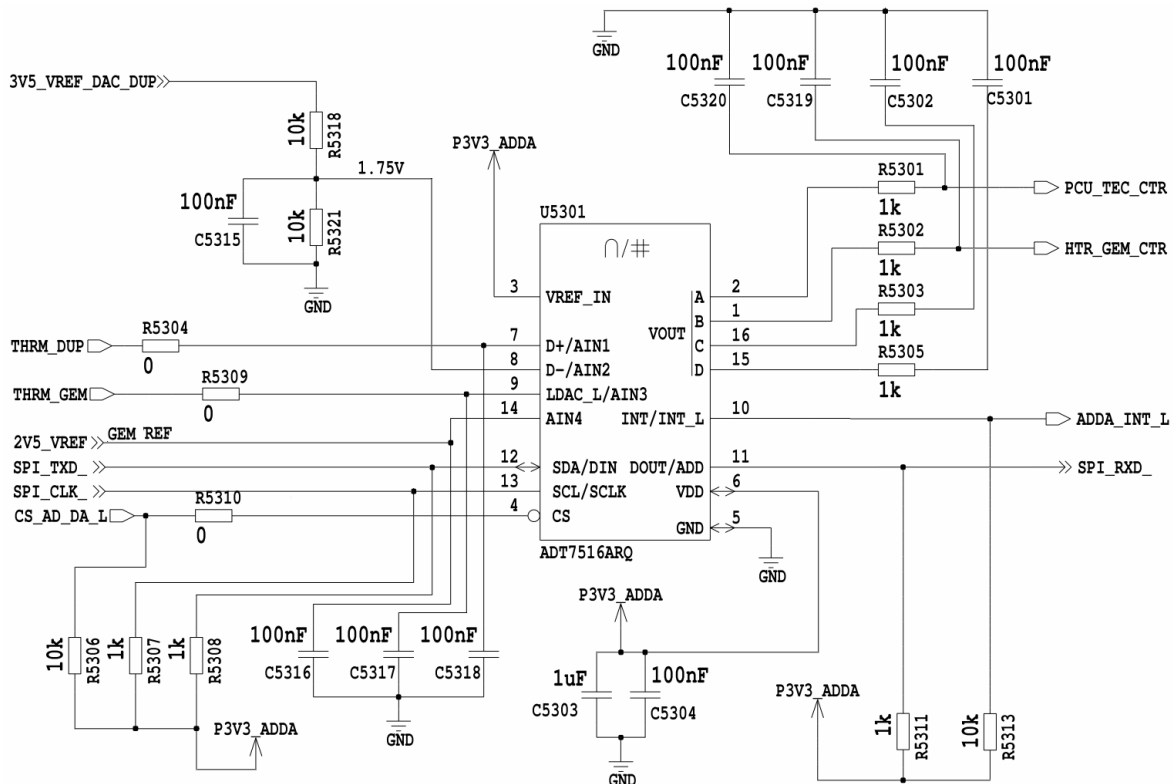


Figura 2.17 – DACs e ADCs para controle da temperatura dos VOAs DUPONT e GEMFIRE

As ADCs são usadas para que o CPU possa obter informação sobre a temperatura a que está o VOA. As tensões de entrada são obtidas a partir do termistor. Apenas uma destas ADCs está ativa em cada O08VA-1. Existem os sinais na entrada das ADCs:

- THRM_DUP – sinal analógico, entrada da ADC, obtido a partir do termistor do VOA DUPONT, que contém informação sobre a temperatura a que está o VOA.
- THRM_GEM – sinal analógico, obtido a partir do termistor do VOA GEMFIRE, que indica a temperatura a que o VOA está.

O ADT7516 possui uma referência interna de 2.25V, que é usada pelas ADCs deste componente. Assim, o *input range* das ADCs é 0V a 2.25V.

As comunicações entre o CPU e o ADT7516 (DACs e ADCs) são efetuadas através do barramento SPI. O barramento SPI usa os sinais SPI_TXD_, SPI_RXD_, SPI_CLK_ e CS_AD_DA_L.

2.6.3.2. Controlo da temperatura do VOA DUPONT

O VOA DUPONT possui um *Thermoelectric Cooler* (TEC), que permite o controlo da sua temperatura.

O TEC é composto por duas placas de cerâmica paralelas, com uma matriz de inúmeras células *Peltier* entre elas – ver Figura 2.18.

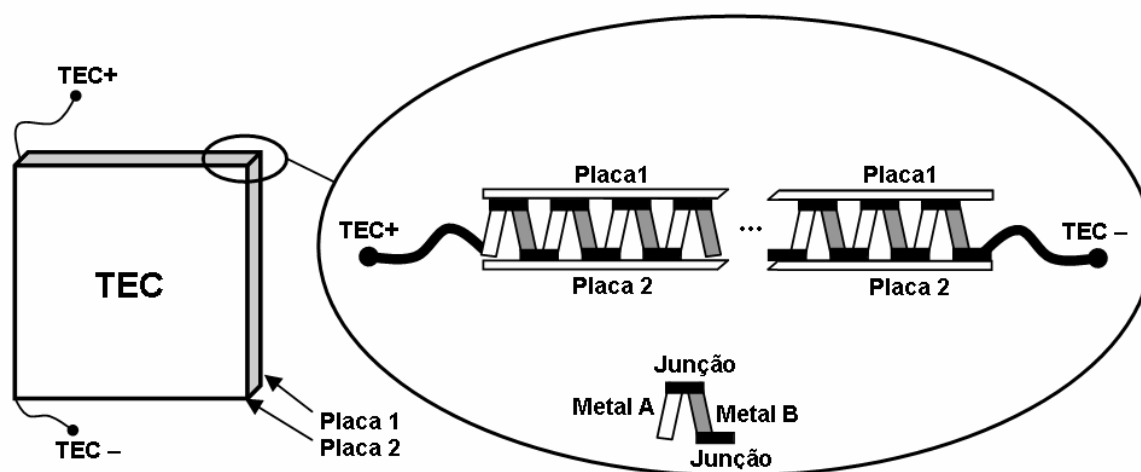


Figura 2.18 – *Thermoelectric Cooler* (TEC)

Uma célula *Peltier* é uma junção de dois metais com características tais que, quando passa corrente em determinado sentido pela junção, esta aquece. Quando a corrente tem sentido inverso, a junção arrefece.

A aplicação de uma tensão aos terminais do TEC provoca corrente através das células *Peltier*.

Quando TEC+ em relação a TEC- é positivo, a corrente flui do metal A para o metal B junto da Placa 1, e do metal B para o metal A junto da Placa 2. Se os metais A e B tiverem características tais que nas suas *junções* existam células *Peltier*, o efeito é que uma das placas aquece e a outra arrefece. Se TEC+ é negativo em relação a TEC-, o efeito é contrário: a placa que arrefecia, aquece, e a placa que aquecia, arrefece.

A aplicação de uma tensão contínua aos terminais do TEC provoca, portanto, uma diferença de temperatura entre as suas placas. Se a temperatura de uma das placas for mantida

aproximadamente estável, através de, por exemplo, um dissipador e uma ventoinha, a outra placa será a que terá a temperatura alterada com a passagem de corrente. Controlando a tensão aplicada ao TEC, consegue-se controlar a temperatura dessa *outra placa* e do que lhe estiver acoplado.

Para se manter estável a temperatura de uma das placas do TEC, usa-se normalmente uma ventoinha junto dessa placa. No caso presente, considerando que o *Rack* onde a O08VA-1 é instalada tem ventoinhas, resolveu-se não instalar nenhum meio adicional. Considerou-se que a ventilação do *Rack* e a dissipação para o chassis (para a *board* da O08VA-1) são suficientes.

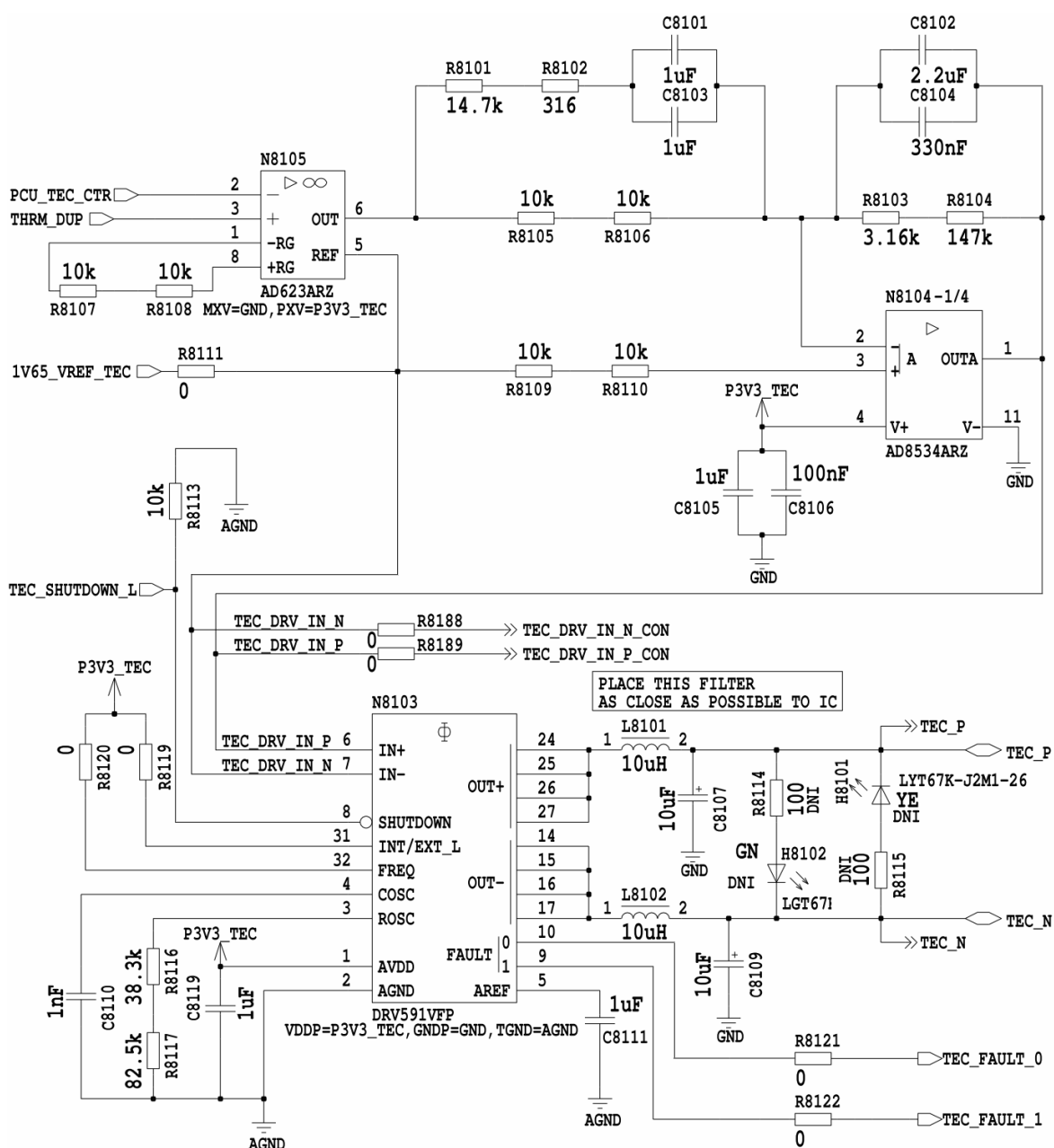


Figura 2.19 – Circuito de controlo do TEC

A Figura 2.19 representa o esquema de controlo do TEC.

Ao amplificador de instrumentação AD623 são aplicados os sinais THRM_DUP (do termistor NTC do VOA) e PCU_TEC_CTR. O sinal PCU_TEC_CTR provém do CPU, através de uma DAC, e informa a temperatura a que o VOA deve funcionar. A resistência entre $-RG$ e $+RG$ define o ganho do amplificador. A referência REF define a tensão de saída, quando THRM_DUP é igual a PCU_TEC_CTR, ou seja o offset.

O OPAMP AD8534 – 1/4, com a sua malha de feedback R-C, adapta o sinal para ser aplicado ao DRV591.

O DRV591 é um *PWM (Pulse-Width Modulation) Power Driver*, adequado para controlo do TEC, que, em função das entradas IN+ e IN-, gera tensão entre as saídas OUT+ e OUT-. Quando a tensão IN+, em relação a IN-, é positiva, o TEC arrefece o VOA. Quando a tensão IN+, em relação a IN-, é negativa, o TEC aquece o VOA.

As saídas TEC_P e TEC_N são aplicadas ao TEC incluído no VOA.

O *Pinout* do VOA DUPONT, com indicação das ligações à Placa, está representado na Figura 2.9.

2.6.3.3. Controlo da temperatura do VOA GEMFIRE

A informação sobre o valor da temperatura a que o VOA deve funcionar – sinal HTR_GEM_CTR – é enviada pelo CPU, através de uma DAC – ver Figura 2.17.

O VOA possui um par de resistências, de 6.4Ω cada, para o aquecer. Possui também um termistor, do tipo NTC, para o controlo da temperatura.

Apenas a possibilidade de aquecimento está implementada neste VOA.

Para que o VOA esteja a uma temperatura constante, a solução é definir uma temperatura que não seja ultrapassada, e aquecer o VOA sempre que a temperatura desça abaixo desse valor.

Segundo o fabricante, 3W são suficientes para aumentar em $30\text{ }^{\circ}\text{C}$ a temperatura do VOA.

A temperatura ideal a que o atenuador deve funcionar (indicada pelo fabricante) é armazenada em memória EEPROM da O08VA-1, no fabrico. Na inicialização da O08VA-1, o CPU envia o valor da temperatura à DAC.

A referência de 2.5V, aplicada à DAC, estabelece o valor máximo da temperatura.

Considerando estes dados, elaborou-se o circuito da Figura 2.20 para controlar a temperatura do VOA.

A informação sobre a temperatura do VOA (sinal THRM_GEM), provinda do termistor, é aplicada à entrada não-inversora do OPAMP N8104-3/4.

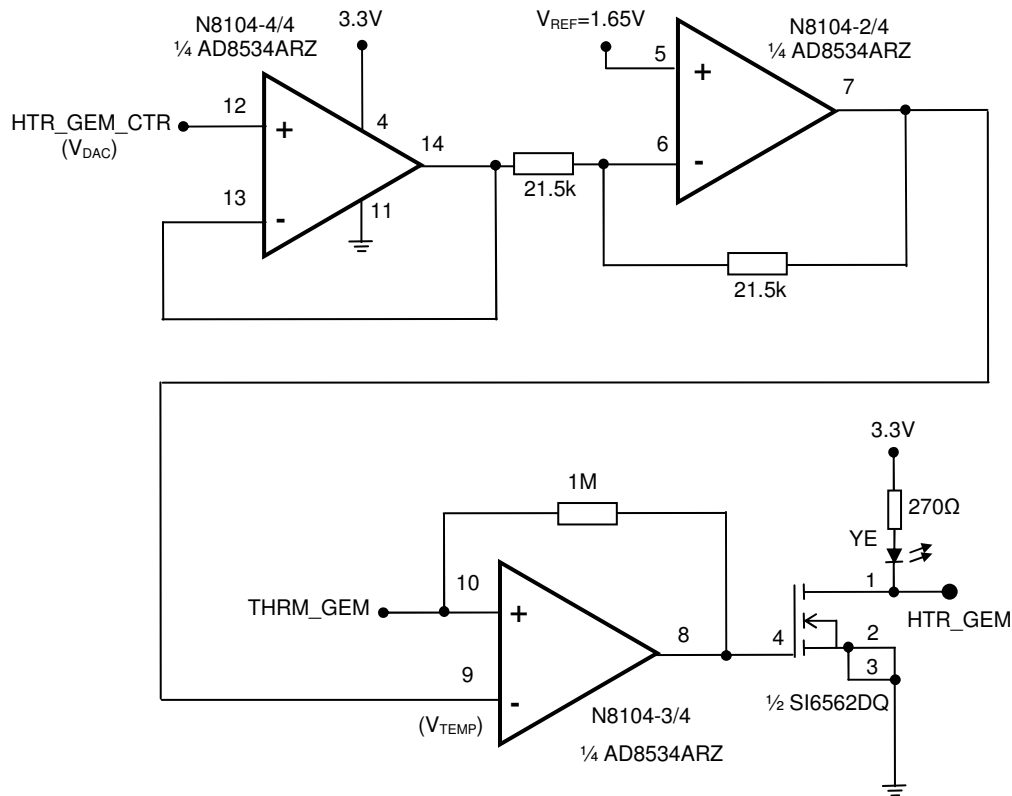


Figura 2.20 – Controlo da temperatura do VOA GEMFIRE

Durante o *Boot* do Sistema e noutras situações, algumas imprevisíveis, as saídas das DACs estão a 0V. Se HTR_GEM_CTR fosse aplicado diretamente à entrada do OPAMP N8104-3/4 o VOA ficaria a aquecer sem controlo. Poderia até provocar o aquecimento excessivo do VOA. A solução para este problema foi adicionar o N8104-2/4, que funciona como um inversor com offset $2 \cdot V_{REF}$. Assim, quando $V_{DAC} = 0V$ a temperatura é a mínima.

N8104-4/4 está configurado como um *voltage follower*, que serve de andar tampão entre a DAC e N8104-2/4.

O software faz as adaptações necessárias ao nível da tensão de saída da DAC, considerando os seguintes dados.

$V_{DAC} = 3.3 - V_{TEMP}$, em que V_{TEMP} (sinal na entrada inversora de N8104-3/4) é equivalente à tensão da NTC, que corresponde à temperatura do VOA.

0V à saída da DAC, corresponde à temperatura mínima.

Tendo a DAC uma referência de 2.5V, V_{TEMP} variará entre 0.8V e 3.3V.

$V_{TEMP} = 3.3V$ corresponde à temperatura mínima, e $V_{TEMP} = 0.8V$ corresponde à temperatura máxima. A tensão V_{TEMP} não poderá ir abaixo de 0.8V, o que corresponde a dizer que o VOA não poderá aquecer a uma temperatura acima da correspondente a 0.8V no termistor (NTC).

O feedback é aplicado na entrada não-inversora do OPAMP N8104-3/4. Mas, como o feedback provém de uma NTC, com um terminal à massa, o sinal de feedback já é negativo, pelo que não necessita de ser invertido pelo OPAMP. Trata-se, pois, de uma configuração de feedback negativo. Por isso, a tensão na NTC tende a aproximar-se de V_{TEMP} . A resistência de 1M serve para adicionar histerese.

O *N-MOSFET* tem uma resistência *ON* muito baixa, pelo que, é aplicada uma potência próxima de 3.4W, para aquecimento.

O LED amarelo acende quando o VOA estiver a ser aquecido.

Uma implementação mais simples do que a representada na Figura 2.20 teria sido possível se o VOA não tivesse um terminal do termistor internamente ligado à massa (GND).

O *Pinout* do VOA GEMFIRE, com indicação das ligações à Placa, está representado na Figura 2.15.

2.6.4. Calibração

Como já foi referido, a O08VA-1 funciona em *loop* aberto e não tem nenhum meio de medição da potência ótica. Por isso, necessita de meios precisos para o controlo da atenuação.

Os VOAs apresentam, de exemplar para exemplar, algumas diferenças, ainda que pequenas, ao nível da relação resistência/temperatura do termistor e da relação controlo/atenuação. São diferenças que afetam a precisão se não forem convenientemente tratadas.

Para resolver este problema, acordou-se com os fabricantes que cada exemplar de VOA seja acompanhado de tabelas para o controlo da temperatura e para o controlo da atenuação.

Estas tabelas são fornecidas em *ficheiros texto*. Estes ficheiros são carregados na memória EEPROM da O08VA-1, no processo de fabrico, onde ficam permanentemente disponíveis.

2.6.4.1. Calibração do VOA DUPONT

Controlo da Temperatura do VOA

No que respeita ao VOA DUPONT, para o controlo do TEC, o fabricante disponibiliza dois ficheiros:

- *R-T ratio.txt* – contém uma tabela da resistência do termistor em função da temperatura do VOA;
- *Temperature.txt* – contém a indicação do valor exato da temperatura a que o VOA deve funcionar.

Estes ficheiros também são usados para detetar situações de excesso de temperatura e gerar alarmes.

A programação da temperatura do VOA é feita na inicialização da O08VA-1. O Software consulta o ficheiro *temperature.txt*, e fica a saber a que temperatura o VOA deve funcionar. Em seguida, consulta o ficheiro *R-T ratio.txt* para saber qual a resistência que o termistor terá a essa temperatura. Em seguida, o CPU calcula, para a resistência do termistor obtida, a tensão aos terminais do termistor, considerando que está em série com uma resistência de 2.61K Ω , que tem uma alimentação de 2.5V – ver Figura 2.9. Em seguida, o CPU envia à DAC o valor da temperatura a que deve funcionar o VOA, aparecendo em PCU_TEC_CTR – ver Figura 2.17 – a tensão correspondente.

Monitorização da Temperatura do VOA

A monitorização da temperatura do VOA é feita sempre que houver disponibilidade do CPU. É uma tarefa de baixa prioridade. O CPU lê a ADC, que contém a tensão correspondente à temperatura do VOA – sinal THRM_DUP – (ver Figura 2.17), consulta a tabela *R-T ratio.txt* para saber a que temperatura está o VOA, e, se a temperatura estiver fora dos limites, gera um alarme ao sistema.

O VOA DUPONT está preparado para funcionar a temperaturas que vão de -5°C a +70°C. Nos testes de Certificação Europeia e Americana, o Produto é submetido a temperaturas que vão desde -5°C a +55°C. O CPU não deve gerar alarmes nestes testes. Assume-se, devido à ação do TEC, que a temperatura do VOA nunca desça abaixo de -5°C. A temperatura pré-definida de funcionamento do VOA (*default temperature*) é 60°C, e o valor final, para cada exemplar, estará próximo de 60°C. Por isso, o limite mínimo e o limite máximo da temperatura de funcionamento do VOA DUPONT foram fixados em -5°C e +65°C, respetivamente.

Controlo da Atenuação

Para o controlo da atenuação, a DUPONT envia, juntamente com cada VOA, um ficheiro com a relação Atenuação/Tensão, para todos os 8 canais, para atenuações de 0 a 20 dB, em *steps* de 0.1 dB.

2.6.4.2. Calibração do VOA GEMFIRE

No que respeita ao VOA GEMFIRE, o fabricante fornece, acompanhado de cada exemplar de VOA, os ficheiros necessários para controlo.

O método é semelhante ao atrás descrito para o VOA DUPONT, com algumas diferenças.

Controlo da Temperatura do VOA

O processo de controlo da temperatura é idêntico ao usado para o VOA DUPONT. Apenas que, nos VOAs GEMFIRE, a temperatura de funcionamento é igual em todos eles. O fabricante fornece os ficheiros necessários.

Monitorização da Temperatura do VOA

O processo de monitorização da temperatura do VOA é idêntico ao usado para o VOA DUPONT. Apenas as temperaturas de funcionamento são diferentes. O VOA GEMFIRE está preparado para funcionar numa larga gama de temperaturas, que vão de -5°C a $+65^{\circ}\text{C}$. Nos testes de Certificação Europeia e Americana, o produto será submetido a temperaturas que vão desde -5°C a $+55^{\circ}\text{C}$. A O08VA-1 não deve gerar alarmes nestes testes. Assume-se, devido à ação do *Substrate Heater*, que a temperatura do VOA nunca desça abaixo de -5°C . Por isso, o limite mínimo e o limite máximo da temperatura de funcionamento do VOA GEMFIRE foram fixados em -5°C e $+65^{\circ}\text{C}$, respetivamente.

Controlo da Atenuação

Para o controlo da atenuação, a GEMFIRE envia, juntamente com cada VOA, um ficheiro com a tabela da atenuação em função da corrente de controlo. Mas, ao contrário da DUPONT, a GEMFIRE não fornece tabelas com *steps* de 0.1dB. Os *steps* são muito maiores. A GEMFIRE sugere que se faça interpolação linear para valores de atenuação intermédios.

2.7. Testes e Resultados

Os primeiros testes efetuados foram testes de rotina, que foram os seguintes:

- Verificação das tensões de alimentação;
- Verificação e registo dos *diagramas de olho*, dos sinais digitais, para se avaliar a qualidade dos sinais e a imunidade ao ruído.

Em seguida foram testadas as funcionalidades da O08VA-1, através da aplicação de lasers de várias potências (0 a 21 dBm), de vários comprimentos de onda (1528 a 1610 nm), nas entradas óticas, com atenuadores variáveis (0 a 20 dB), e medida a potência de saída do laser. Os resultados mostraram-se de acordo com as especificações expressas na Tabela 2.6.

Testou-se, também, a estabilidade (*Attenuation drift*), ao longo de alguns dias de funcionamento contínuo; e a *reproducibility* em situações de *Cold Starts* sucessivos da O08VA-1, tendo-se verificado, nestes aspetos, cumprir os *Requisitos Óticos* especificados na Tabela 2.1.

Em relação ao controlo da temperatura dos VOAs, verificou-se o normal funcionamento.

Terminado o desenvolvimento, a O08VA-1 foi submetida a testes de Certificação Europeia (CE). Numa primeira fase o produto foi submetido à *Autoridade Nacional de Comunicações (ANACOM)*, para testes preliminares.

Seguiram-se testes mais intensos e rigorosos realizados na Alemanha. Durante semanas, o produto foi testado em diversas áreas, sendo submetido às condições mais extremas, e testado o seu normal funcionamento.

Os tipos de testes que se fizeram, para obter as *certificações* respetivas, foram:

- Testes em plataforma vibratória para testes de robustez mecânica, imunidade vibratória
- Testes de temperatura e humidade
- Testes de radiações eletromagnéticas
 - Ruído emitido pelo produto
 - Imunidade a ruído submetido ao produto
- Testes de ruído na alimentação
 - Ruído emitido pelo produto
 - Imunidade a ruído na alimentação
- Testes de imunidade a descargas eletrostáticas (eram aplicadas descargas de alguns Kilovolts na caixa do produto).

Obteve-se um módulo que permite controlar a potência de 8 canais de laser. Inserindo M destes módulos no sistema, fica-se com $N=M \times 8$ atenuadores óticos – ver Figura 2.21.

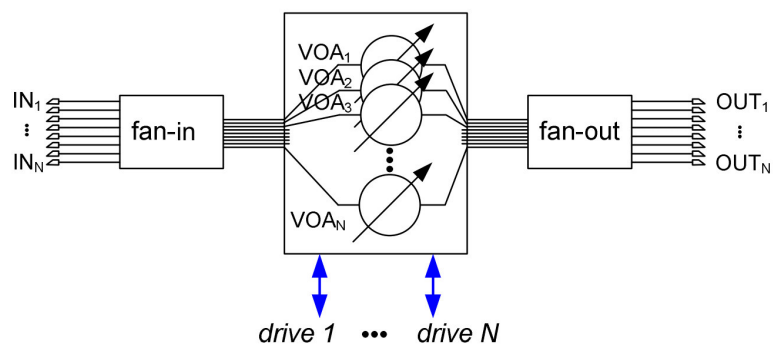


Figura 2.21 – Configuração ótica de um agrupamento de N atenuadores

2.8. Bibliografia

- Taub, H., & D. Schilling, *Digital Integrated Electronics*
McGRAW-HILL
- Taub, H., *Digital Circuits and Microprocessors*
McGRAW-HILL
- Jacob Millman & Arvin Grabel, *Microelectronics*
© 1988 McGRAW-HILL
- Documentação interna da empresa *Siemens*
- Paul R. Gray & Robert G. Meyer, *Analysis and Design of Analog Integrated Circuits*
© 1984 John Wiley & Sons, Inc.

Capítulo 3

Gestão de Redes

3.1. Local e período em que decorreu

Na empresa *Siemens, SA*, nos anos de 2002 e 2003, o autor trabalhou no desenvolvimento de software para *Gestão* de equipamentos de comunicações – *Gestão de Redes (Network Management)*.

3.2. Introdução

Este projeto surgiu integrado numa linha de desenvolvimento de equipamentos para redes de comunicações direccionados para Operadores de Telecomunicações. Consistiu no desenvolvimento de *Gateways*.

Gateways são equipamentos de telecomunicações que possuem várias e diversas interfaces – ver Figura 3.1. Necessitam de ser configuradas, monitorizadas, auditadas e controladas, tanto na fase de desenvolvimento como no seu uso subsequente. Uma forma de o fazer seria construir um

sistema proprietário. No entanto, isso não seria nada prático nem economicamente viável. A forma mais adequada para efetuar o controlo dos equipamentos de rede é utilizar a própria rede onde o equipamento opera.

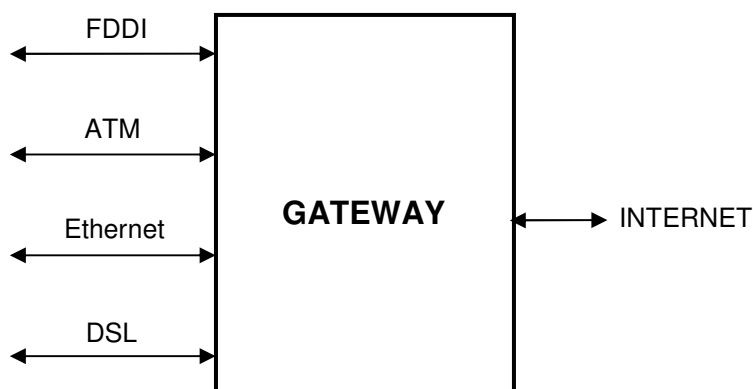


Figura 3.1 – Gateway

Atualmente, as redes TCP/IP são as mais comuns. Um protocolo padrão a que se chamou *Simple Network Management Protocol* (SNMP), que funciona em redes TCP/IP, permite efetuar o controlo dos equipamentos de rede (e da rede propriamente dita) – a *gestão da rede* (Network Management).

Uma vasta equipa distribuída pela Alemanha, Portugal e Índia, desenvolveu estes equipamentos de telecomunicações. Em Portugal estava localizado o desenvolvimento de parte do software para *Gestão de Redes*.

3.3. Objetivos e especificações

Na *Siemens* os projetos estavam funcionalmente divididos por zonas. Em Portugal estava sediado algum do desenvolvimento de software que possibilitasse a *Gestão das Redes*.

Uma grande parte deste software aplicava-se a todos os projetos desta natureza. A principal diferença entre os projetos estava no equipamento que se desenvolvia; aspetos específicos de performance, velocidade, tipo de interfaces distinguiam-nos.

Ao nível da *gestão de redes* uma parte do software era padrão, sendo outra mais específica, com elementos definidos pela empresa.

Em concreto, o software de *gestão de redes* desenvolvido neste projeto tem as seguintes capacidades:

- Configurar a *Gateway* – configurar cada uma das interfaces deste equipamento, bem como as respectivas comunicações;
- Monitorizar o desempenho da *Gateway*, assim como da rede – informar sobre a velocidade e sucesso das transferências de dados;
- Detetar falhas de rede ou acesso inadequado – no caso de ocorrerem os seguintes eventos na *Gateway*, o sistema de gestão informa a situação:
 - Acesso inadequado;
 - Ativação de interface;
 - Desativação de interface;
 - Arranque da *Gateway* a frio;
 - Arranque da *Gateway* a quente;
- Auditar a utilização da rede – monitorizar a utilização global da rede, informando sobre o volume de dados transferidos e estatísticas de utilização dos recursos;
- Implementar meios para garantir a segurança e confidencialidade das comunicações.

3.4. Responsabilidades e competências específicas

Neste projeto o autor desempenhou as funções de *Software Developer*. Desenvolveu software para *gestão de redes*, usando as linguagens JAVA e XML. Adicionalmente testou e assegurou a manutenção do software.

3.5. Enquadramento teórico

O projeto aqui reportado foi desenvolvido numa altura em que as comunicações de dados estavam a deixar os meios mais tradicionais (POTS, RDIS), e estavam a passar para ADSL, Cabo, Sem Fios, Fibra Ótica.

As *Gateways* são equipamentos para esse fim – possuem interfaces diversas, que permitem a comunicação entre vários tipos de equipamentos.

As *Gateways* de Operador de Telecomunicações necessitam meios para o controlo das comunicações nas suas interfaces. O Protocolo SNMP é adequado para esse fim. Corre sobre o TCP/IP, o qual interliga diversos tipos de equipamentos. Ao estar instalado sobre o TCP/IP, o SNMP permite efetuar a *Gestão da Rede* de diversos tipos de sistemas.

De um modo geral, as redes de telecomunicações necessitam de acompanhamento permanente. São assuntos críticos, que necessitam de meios para:

- Configurar os equipamentos que estão na rede;
- Monitorizar o funcionamento de rede;
- Detetar e sinalizar falhas na rede ou acesso inadequado;
- Auditar a utilização da rede;
- Segurança da rede.

Com software auxiliar de *Gestão de Redes* adequado, o SNMP permite efetuar todas estas operações.

O *agente de SNMP*, que é o software de SNMP que se instala no equipamento gerido – *Gateway* – controla, não só a *Gateway*, mas também todo o tráfego que nela circula, que é o tráfego de rede. Um *sistema de gestão de SNMP* envia-lhe instruções e recebe informações para a *gestão* adequada da rede.

O SNMP permite não só efetuar operações padrão (tais como criar e configurar camadas do Modelo OSI numa interface) como também operações específicas definidas pelo fabricante. Segue-se uma breve descrição teórica do SNMP.

3.5.1. SNMP

O SNMP é instalado em redes TCP/IP e permite efetuar, sobre os equipamentos de rede, a que chama *agentes*, operações de configuração, monitorização e manutenção.

Num sistema SNMP existe um ou mais *gestores* SNMP e um ou mais *agentes* SNMP. O *gestor* SNMP efetua operações de SNMP sobre o(s) *agente(s)*.

O *gestor de SNMP* é, tipicamente, uma aplicação de *gestão de rede* a correr num computador convencional, que está na rede, com, por exemplo, o sistema operativo *Windows* e o SNMP instalado.

Um *agente de SNMP* é uma aplicação instalada, em geral, noutro equipamento que não aquele que efetua a *gestão da rede*, e que pode ser um router, uma *gateway*, outro computador ou qualquer outro tipo de equipamento que tenha instalado o SNMP.

Para que o SNMP possa ser utilizado tem de estar *iniciado*.

3.5.1.1. Mensagens e Traps

Para efetuar a *gestão da rede*, o *gestor* SNMP (SNMP Management System) envia *mensagens* (*Requests*) ao *agente de SNMP*. Estas mensagens são do tipo *Get*, *GetNext*, *GetBulk*, *Set* – ver Figura 3.2.

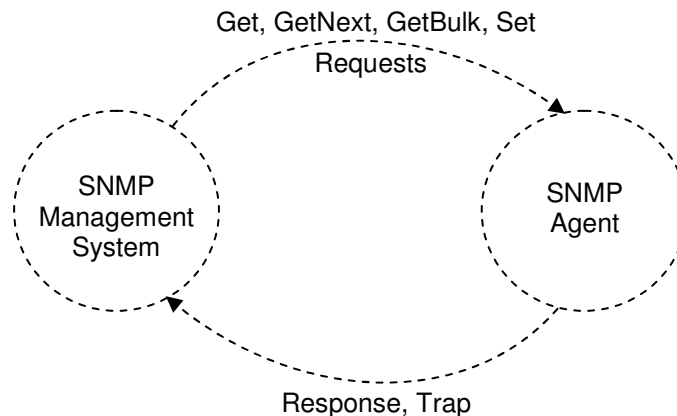


Figura 3.2 – O sistema SNMP

No *agente de SNMP* existe uma base de dados, chamada MIB (*Management Information Base*), que contém os *Objetos* (*variáveis*) das *mensagens*. No Apêndice A, apresentam-se alguns excertos da RFC1213-MIB, que contém as *variáveis* que são a base do SNMP. A *RFC1213-MIB* é uma MIB, presente em todos os sistemas SNMP, que contém informações sobre a rede, e, além disso, permite efetuar algumas operações de configuração da mesma rede.

O *agente de SNMP*, ao receber uma *mensagem* do *gestor de SNMP*, age em conformidade. Se se tratar de uma *mensagem* do tipo *Set*, o *agente* executa os procedimentos associados ao *Objeto*, – por exemplo, ativar uma interface – e altera o valor da *variável* na MIB. Finalmente, o *agente* responde ao *gestor*, informando-o do resultado da operação. A *mensagem Set* permite, assim, efetuar alterações no *agente*.

Se se tratar de uma *mensagem* do tipo *Get*, *GetNext*, *GetBulk*, o *agente* consulta a MIB, e responde ao *gestor*, informando-o do valor que a *variável* tem.

Nas *mensagens* SNMP, são especificados o endereço IP do *agente*, a *variável* da MIB e, se se tratar de uma *mensagem Set*, o valor que se pretende que a *variável* assuma.

No geral, o *agente de SNMP* só responde a *mensagens* do *gestor*. Mas, quando ocorre um *Evento de Excepção* no *agente*, este, espontaneamente, envia uma *Trap*, em geral ao *gestor*, para o informar do *Evento*. No SNMP Service do *agente*, existe um conversor *Evento – Trap*. No SNMP Service do *gestor*, existe um conversor *Trap – Evento*. Por exemplo, uma situação de *Trap* é a seguinte, extraída da RFC1213-MIB:

```
linkDown TRAP-TYPE
    ENTERPRISE    snmp
    VARIABLES     { ifIndex }
    DESCRIPTION   "A linkDown trap signifies that the sending
                  protocol entity recognizes a failure in one of
                  the communication links represented in the
                  agent's configuration."
```

O *agente* envia a *Trap linkDown* quando o equipamento em que está instalado desativa uma interface, em resultado de uma situação de excepção; ***linkUp***, ***coldStart***, ***warmStart***, ***authenticationFailure*** são outros exemplos de *Traps*, também da *RFC1213-MIB* – ver Apêndice A.

Os destinos das *Traps* (um *agente* pode enviar *Traps* a mais do que um *host*) podem ser outros que não o *gestor* SNMP que faz a *gestão*. No *SNMP Service* do *agente*, são especificados os destinos das *Traps*. Podem ser quaisquer *hosts* que estejam na rede e que tenham o *SNMP Service* iniciado.

3.5.1.2. Base de dados MIB

O *agente de SNMP* possui uma base de dados SNMP, a que se chama MIB (*Management Information Base*). A MIB contém um conjunto de *Objetos*, mais conhecidos por *variáveis*. As *variáveis* estão agrupadas em ficheiros a que também se chamam MIBs. Por exemplo, existe a *RFC1213-MIB* (ver Apêndice A), a *RADIUS-ACC-SERVER-MIB*.

Existe um conjunto de MIBs padrão, internacionalmente definidas, como as atrás referidas. No entanto, o fabricante do equipamento de rede pode criar MIBs próprias, para executar operações específicas no seu equipamento.

Cada uma das MIBs possui um conjunto de *Objetos* (*variáveis*) associados a uma determinada função. Por exemplo, a *RFC1213-MIB* está associada ao TCP/IP, e contém a lista e características das interfaces, as tabelas de *routing*, estatísticas sobre a atividade da rede, etc.

As *variáveis* da MIB estão estruturadas em árvore. Numa *mensagem* SNMP, a *variável* pode ser especificada pelo seu nome ou pelo OID (*Object Identifier*). O OID é uma sequência de números separados por pontos. De ramo para ramo, são acrescentados um ponto e um número ao OID.

As operações que o *gestor* SNMP executa (através do *agente*) sobre a MIB são de leitura (*Get*, *GetNext*, *GetBulk*) e escrita (*Set*).

Uma *variável* da MIB tem, tipicamente, as seguintes especificações:

- Nome
- Sintaxe – tipo de *variável* (*integer*, *string*, *counter*, *gauge*, *TimeTicks*, etc)
- Acesso – o tipo de acesso que é permitido (*read-only*, *read-write*, *not-accessible*)

- Estado – (*mandatory, current, deprecated*)
- Descrição – possui uma pequena descrição textual do significado da *variável*.

Um exemplo de *variável* da *RFC1213-MIB*, com apenas permissão de leitura, é o seguinte:

```
ipInReceives OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of input datagrams received from
        interfaces, including those received in error."
    ::= { ip 3 }
```

Um exemplo de *variável* da *RFC1213-MIB*, com permissão de leitura e escrita, é o seguinte:

```
ipDefaultTTL OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The default value inserted into the Time-To-Live
        field of the IP header of datagrams originated at
        this entity, whenever a TTL value is not supplied
        by the transport layer protocol."
    ::= { ip 2 }
```

3.5.1.3. Segurança

A *segurança* do SNMP é garantida através da utilização de nomes de *Comunidades*, *Traps* de autenticação e restrições ao nível do tipo de acesso à MIB.

Os mecanismos de *segurança* são implementados do lado do *agente de SNMP*. O *agente* pode limitar o acesso dos *gestores de SNMP* à MIB.

As *Comunidades* são definidas como grupos de *hosts* na rede, com acesso à MIB e atuam como que *passwords* para aceder ao *agente*. Um *agente* que tenha uma *Comunidade* que seja *public*, permite o acesso à MIB de qualquer *sistema de gestão de SNMP* que esteja na rede.

Um *agente* que receba uma *mensagem* de um *gestor SNMP* que não pertença à lista (i.e. à *comunidade*) de *gestores* dos quais são aceites *mensagens*, rejeitará a *mensagem* e enviará uma *Trap de Autenticação*.

Mecanismos de acesso seguros podem também ser implementados especificando uma lista de *gestores de SNMP* dos quais são aceites *mensagens*. É uma alternativa (complemento) ao uso de *Comunidades*.

3.6. Projeto e implementação

3.6.1. Introdução

No período em que o autor esteve envolvido neste projeto, o desenvolvimento de equipamentos de comunicações na *Siemens* estava geograficamente distribuído pela Alemanha, Portugal e Índia – ver Figura 3.3.

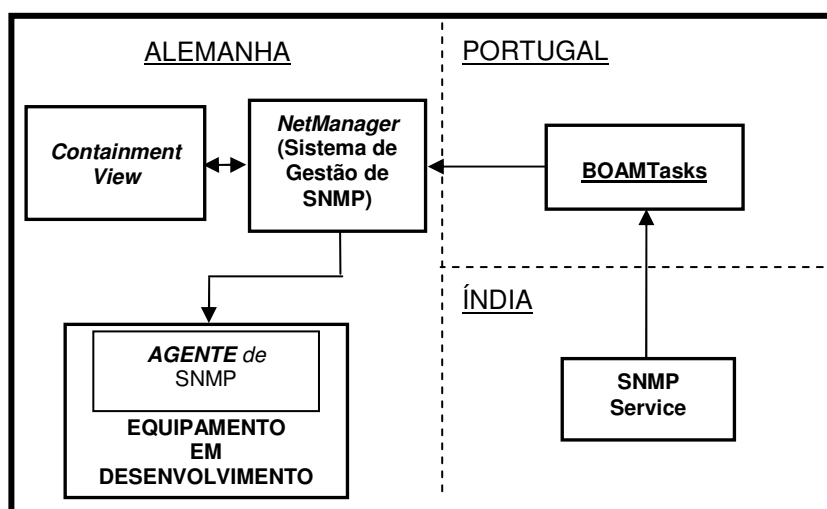


Figura 3.3 – Gestão de Redes – as equipas de desenvolvimento

Na Alemanha estava o *Project Leader*, a coordenação, o equipamento em desenvolvimento e todo o hardware de destino do software.

Em Portugal estava localizado o desenvolvimento do código para fazer operações de *Gestão de Rede* através do SNMP – as *BOAMTasks* (*Basic Operation and Maintenance Tasks*).

Na Índia desenvolviam-se as rotinas de SNMP que eram usadas pelas *BOAMTasks*.

3.6.2. As BOAMTasks

3.6.2.1. BOAMTasks – estrutura

Do lado da equipa que desenvolvia as BOAMTasks, o projeto iniciava-se com a análise das MIBs, que eram enviadas pela equipa do hardware. Esta última elaborava a lista de operações para *gestão da rede* (lista de BOAMTasks), a efetuar sobre o *agente*.

As BOAMTasks efetuam “simples” operações de SNMP (*Set*, *Get*, *GetNext*, *GetBulk*), sobre o *agente*. São pequenos programas em *Java*, com cerca de 15 linhas cada, que efetuam, tipicamente, uma operação para *Gestão da Rede*.

As *tasks* têm uma entrada (*input*) e algumas, saídas (*output*) – Figura 3.4. O *input* (*input list*) e *output* (*output list*) são especificados à parte em ficheiros XML (*Extensible Markup Language*). Também, nestes ficheiros, era incluído *help text* para descrever qual a finalidade da *Task* e o significado dos parâmetros de *input*. O XML é especialmente adequado para especificação de dados estruturados. Por exemplo, a *Task CreateIProute* tem os ficheiros: *CreateIProute.java* e *CreateIProute.inp*. A *Task DisplayIProute* tem os ficheiros *DisplayIProute.java*, *DisplayIProute.inp* e *DisplayIProute.out*.

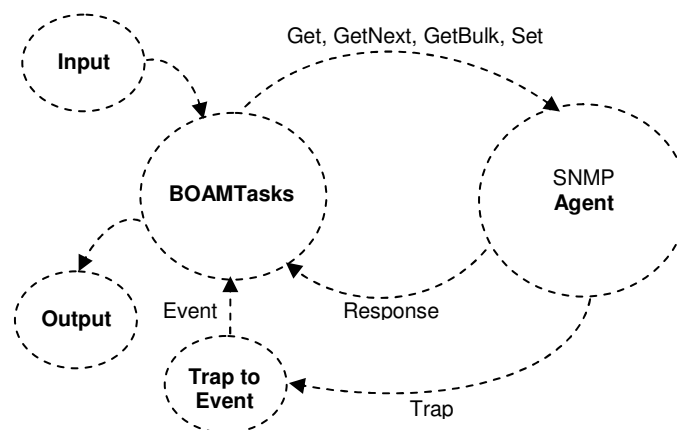


Figura 3.4 – As BOAMTasks na gestão da Rede

No final da *Task* são processados os *Eventos de Exceção* (oriundos de *Traps*), e feito o *output* de uma *mensagem* que esclareça a situação de Exceção.

As operações de *Gestão de Rede* são reunidas em grupos, cada um dos quais pode ter *tasks* de *Create*, de *Display*, de *Modify* e de *Delete*. Por exemplo: *CreateIProute*, *DisplayIProute*, ...

3.6.2.2. BOAMTasks na gestão da rede

A MIB-II, que é uma evolução da MIB-I, contém os *objetos* (*variáveis*) que estão na base da *gestão de redes* TCP/IP. A empresa usava a *RFC1213-MIB* – ver Apêndice A – que é a forma (o nome) como a MIB-II é publicada.

A MIB-II contém a informação necessária para se efetuar a monitorização e auditoria da rede. Contém também os elementos (*Objetos*) necessários para se efetuar a configuração base da rede ao nível da camada 3 do Modelo OSI.

Para a configuração de interfaces do tipo ATM (IP over ATM) e FDDI, são usadas MIBs específicas. Construíram-se *tasks* para a gestão dessas MIBs.

A MIB-II contém 11 grupos de *variáveis*, dos quais 7 se usaram para a *gestão* SNMP. Assim, os *grupos* e subgrupos de *objetos* da MIB-II, para os quais se construíram *tasks*, foram os seguintes:

- System group
- Interfaces group
 - Interfaces table
- IP group
 - IP address table
 - IP routing table
 - IP Address Translation table
 - additional IP objects
- ICMP group
- TCP group
 - TCP Connection table
 - additional TCP objects
- UDP group
 - UDP Listener table

O *System group* contém a identificação, descrição, características técnicas e informação sobre o tempo de uso do hardware onde o *agente* está instalado. Para este grupo, construíram-se *tasks* para mostrar e alterar o valor das *variáveis*.

O *Interfaces group* contém a tabela de interfaces. Esta tabela possui a lista e características das interfaces do hardware, assim como dados sobre a atividade de cada interface.

Esta tabela contém uma única variável que pode ser alterada, que é *ifAdminStatus*, que permite ativar e desativar uma determinada interface.

Para este *grupo*, construíram-se *tasks* para mostrar a tabela e ativar e desativar interfaces.

O *IP group* contém vasta informação sobre a atividade da rede e permite efetuar configurações ao nível do encaminhamento de dados.

Para a *IP Address table (read-only)*, construiu-se uma *Task* para a mostrar. Para a *IP routing table*, construíram-se *tasks* para criar, mostrar, modificar e apagar *IP routes*. Para a tabela *IP Address Translation table (read-only)* existe uma *Task* para mostrar a equivalência entre endereços IP e físicos.

O *ICMP (Internet Control Message Protocol) group* tem apenas variáveis *read-only* que podem ser consultadas através da *Task DisplayICMPstatistics*. O grupo ICMP reporta erros e permite uma conectividade simples. O ICMP é utilizado pela ferramenta *Ping* para a resolução de problemas com o TCP/IP.

O *TCP group* contém informação sobre as ligações TCP. A única variável deste grupo que pode ser alterada é *tcpConnState*. Esta variável pode ser alterada apenas para o valor 12, que corresponde a terminar a ligação. Para este grupo, construíram-se *tasks* para mostrar o valor das variáveis e para terminar uma ligação TCP.

O *UDP (User Datagram Protocol) group*, que é um complemento do TCP, contém apenas variáveis *read-only*. Para este grupo, construíram-se *tasks* para mostrar estatísticas UDP e para a tabela UDP.

O *SNMP group* contém informações sobre a atividade do *sistema de gestão SNMP*. As variáveis deste grupo que têm *acesso de escrita e/ou de leitura* não tem interesse para a *gestão da rede*.

Um exemplo de *Task* do grupo IP que foi desenvolvida para mudar o estado de uma das Interfaces é *ModifyIPConfig*. Esta *Task* tem a seguinte configuração.

- A *Task* tem como parâmetros de entrada o endereço *IP* da interface e *ifAdminStatus*.
ifAdminStatus é uma variável da *RFC1213-MIB*, da *Interfaces Table*, que pode tomar os valores 1 (*up*), 2 (*down*) ou 3 (*testing*).
- A *Task* começa por obter o *ifIndex* (é o *Interface Index*), que é a *chave* da Tabela *Interfaces Table*, a partir do *IP Address*:

```
ifIndex = getIfIndex (IPAddress);
```
- Em seguida, efetua o *Set* da variável *ifAdminStatus*, de acordo com o *input* especificado.
- Finalmente, a *Task* processa os eventos (oriundos de *Traps*).

Note-se que apenas a variável *ifAdminStatus* da *Interfaces Table* tem *acesso* do tipo *read-write*. Todas as outras variáveis da *Interfaces Table* são *read-only* ou *not-accessible*. Por isso, a variável *ifAdminStatus* pode ser modificada com operações SNMP.

ModifyIPConfig não modifica as características das Interfaces. As características das Interfaces são definidas pelo hardware. O que a *Task* faz é ativar e desativar as Interfaces.

3.6.2.3. Tasks para segurança das comunicações – Tunneling

Para garantir a segurança da informação construíram-se *tasks* que criam túneis sobre os pacotes IP.

Estes túneis encapsulam os pacotes IP transportados da origem para o destino. Os pacotes encapsulados atravessam a rede dentro do túnel.

Por si só, os túneis não oferecem muita segurança. No entanto, quando o pacote IP encapsulado é encriptado e comprimido, os túneis oferecem elevada segurança quanto à confidencialidade da informação. Por ser encapsulado, o pacote IP pode ser submetido a todo o tipo de encriptação, desde os tipos padrão, até à encriptação personalizada.

A Figura 3.5, representa o modo de túnel *ESP* (*Encapsulating Security Payload*).

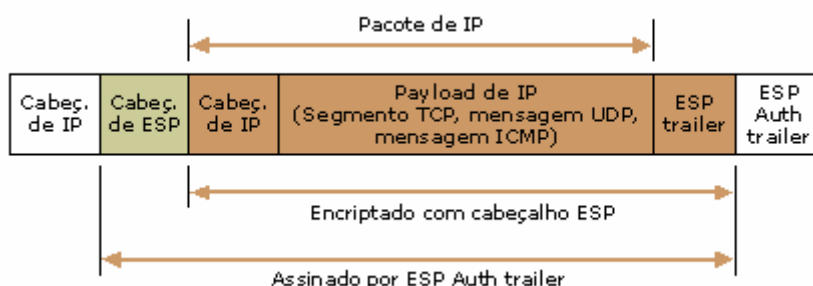


Figura 3.5 – Túnel *ESP*

O pacote IP que tem o *Payload* foi encriptado e encapsulado, por técnicas de ESP, dando origem a um novo Pacote – o *Pacote de Túnel*, com novo cabeçalho.

Uma extremidade do túnel é o local onde o *Pacote de Payload* é encapsulado. A outra extremidade é onde se desencapsula o *Pacote de Payload* ou seja, onde este é extraído.

Uma das aplicações dos túneis é para criar uma VPN (*Virtual Private Network*). Uma VPN pode estar espalhada por qualquer zona do globo, ligada, de forma segura, utilizando redes públicas ou privadas com túneis.

Na Figura 3.6 apresenta-se um exemplo de VPN implementado sobre o L2TP (*Layer Two Tunneling Protocol*) / IPSec.

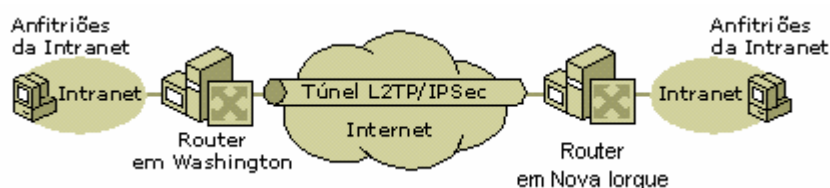


Figura 3.6 – Túnel L2TP/IPSec

O L2TP é utilizado para transmitir dados por túnel através de uma rede partilhada ou pública, tal como a Internet, e o ESP de IPSec é utilizado para encriptar os dados.

Construíram-se *tasks* para criar VPNs seguras sobre túneis L2TP e PPTP *gateway* a *gateway*, e entre utilizador e *gateway*.

A utilização de túneis é transparente para a rede que é utilizada para tráfego.

Com a combinação do L2TP com protocolos de autenticação e encriptação, como o IPSec, a transferência de dados através de uma VPN é bastante segura.

3.6.2.4. As BOAMTasks no Sistema

As BOAMTasks aparecem no *NetManager* como comandos., o que permite estruturar e automatizar a sua utilização.

3.6.3. Aplicações para gestão da rede

O Sistema final é constituído pelo *NetManager*, o *Containment View* e o(s) *Agent(s)* - Figura 3.7.

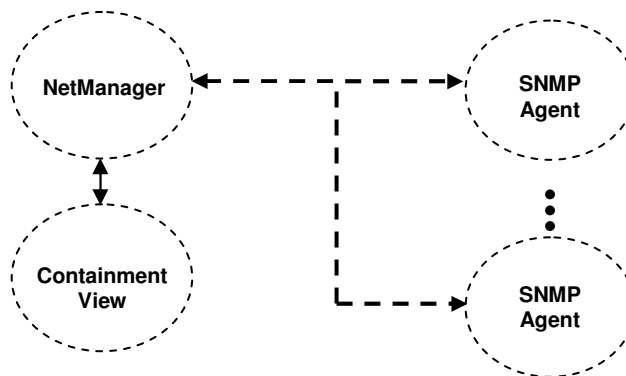


Figura 3.7 – O Sistema final

3.6.3.1. O NetManager

O *NetManager* é uma aplicação que tem por função efetuar a *gestão da rede*. As BOAMTasks são inseridas no *NetManager* e aparecem ao utilizador em lista. Clicando numa *task* aparecem os parâmetros de *input* para serem especificados. Após a especificação do *input*, a *task* é executada e feito o *display* do output.

As *tasks* também podem ser executadas no *NetManager* sem intervenção direta do utilizador. Uma *batch file*, com uma lista de *BOAMTasks* a executar em série, pode ser especificada e corrida. Este procedimento é especialmente útil e necessário no arranque e configuração do equipamento.

O *NetManager* fornece ao *Containment View* informação sobre o estado do equipamento e das interfaces da rede. Além disso recebe do *Containment View* instruções para que envie *mensagens* SNMP ao *agente de SNMP* (operações de *gestão da rede*).

3.6.3.2. O Containment View

O *Containment View* é a interface do sistema com o operador humano. Trata-se de uma aplicação que exhibe permanentemente a lista de equipamentos na rede, respectivas interfaces, e o seu estado (*up/down*). É o meio para monitorizar a rede.

No *Containment View* também se pode fazer a manutenção. Clicando com a tecla direita do rato sobre um equipamento ou interface, e seleccionando a operação pretendida, pode alterar-se o estado do item. O *Containment View* envia a informação necessária ao *NetManager*, que, por sua vez, executa uma ou mais *BOAMTasks*, e *mensagens* SNMP são enviadas ao *agente*, no sentido de executar a operação

3.7. Resultados

O resultado deste projeto foi um pacote de software para *gestão de rede* capaz de ser usado nos principais tipos de redes. Durante a sua utilização o software mostrou ser robusto e praticamente não necessitou de manutenção.

3.8. Bibliografia

- Learning Tree International, *TCP/IP – HANDS ON*
- Philips Omnicom, *Modern Data Networks*
- Microsoft , *Windows XP*
© 1985-2001 Microsoft Corporation

- Sun Microsystems, *Java Programming Manual*

Os *snapshots* aqui apresentados foram extraídos do *Microsoft Windows XP*. Este sistema operativo, incluindo o respetivo *serviço SNMP*, foi o usado neste projeto para a *gestão da rede*.

Capítulo 4

Software para a Rede Telefónica Fixa

4.1. Local e período em que decorreu

Este trabalho decorreu na empresa *Siemens SA*, Amadora, de Abril de 1999 a finais de 2001.

4.2. Introdução

As linhas telefónicas das redes PSTN (*Public Switched Telephone Network*) e ISDN (*Integrated Services Digital Network*), principalmente por estarem no exterior, estão sujeitas a avarias frequentes. O mesmo se passa com os aparelhos telefónicos.

Como a deslocação de um técnico ao local para reparação de avarias é sempre dispendiosa, interessa, pois, ter um meio de diagnóstico das linhas e dos telefones, a partir da Central de Telecomunicações.

O sistema que aqui se descreve torna isso possível, ou seja o diagnóstico remoto das ligações a equipamentos e até dos próprios equipamentos, evitando assim a necessidade de deslocação/intervenção de um técnico no local.

4.3. Objetivos e especificações

Pretende-se que, a partir da Central de Telecomunicações, seja possível fazer o diagnóstico através de comandos MML (*Man-Machine Language*) de linhas telefónicas e telefones, sem a necessidade de deslocação de técnico ao exterior.

Em concreto pretende-se ter a possibilidade de, a partir da Central, efetuar os seguintes testes às linhas telefónicas e aos telefones:

- Medição de tensões na linha telefónica, com o auscultador do telefone *on-hook* e *off-hook*;
- Medição da resistência da linha, com o auscultador do telefone *on-hook* e *off-hook*;
- Medição de capacidades, com o auscultador do telefone *on-hook* e *off-hook*;
- Teste da campainha do telefone;
- Teste de marcação de números de telefone (Pulse e DTMF);
- Teste de estabelecimento de chamadas;
- Teste de tons de telefone.

4.4. Responsabilidades e competências específicas

Neste projeto o autor desempenhou funções de *Software Developer*, desenvolvendo programas, em linguagem *Assembly*, para teste e manutenção de linhas telefónicas e telefones da rede fixa pública, para os clientes da empresa onde trabalhava, que eram companhias de telefones em países espalhados por todo o globo.

Aquilo em que o autor mais trabalhou, enquanto esteve nestas funções, foi em adaptação de software para diversos países clientes da empresa – *country adaptation*. Quando era lançada nova versão de software para as centrais telefónicas, era, primeiramente, feita a versão base, a que a empresa chamava WSA (*World Standard Application*), a partir da qual se faziam as adaptações para cada país. A *country adaptation* englobava os seguintes itens:

- Tons do telefone;

- Tensões da linha telefónica;
- Frequência da tensão da campainha.

A este trabalho de adaptação de software, seguiam-se testes, não só do que foi adaptado, mas também de compatibilidades.

As adaptações são apenas necessárias para linhas telefónicas analógicas. O *ISDN* é um padrão internacionalmente definido, pelo que não tem especificidades nacionais.

O autor integrou a equipa que desenvolveu a versão POR V12 – versão 12, da *Siemens*, para a rede telefónica fixa portuguesa.

No domínio do software em que trabalhou, o autor foi responsável pelos subsistemas JGAMD e JTLMD. Ser responsável por um subsistema, implica que se assegure o seu bom funcionamento em todo o mundo, se efetue a sua manutenção, se disponibilize documentação, se esclareça alguma dúvida que possa existir sobre o mesmo. Sempre que é lançada nova versão de WSA, o responsável pelo subsistema tem de efetuar as alterações necessárias, testes, e assegurar que continua a funcionar corretamente, eventualmente com novas funcionalidades.

4.5. Enquadramento teórico

As linhas telefónicas analógicas PSTN possuem determinadas características elétricas definidas a nível nacional.

Com o auscultador do telefone poitado (*on-hook*), a linha telefónica apresenta-se como um circuito aberto. Em termos de *AC*, apresenta uma determinada capacidade – no telefone há um condensador em série com uma campainha aos terminais da linha telefónica – ver Figura 4.1.

Com o auscultador do telefone levantado (*off-hook*), o telefone aplica, à linha, uma carga com uma determinada impedância.

Se a linha tiver uma tensão DC aplicada, provinda de fonte com resistência interna, não deve haver queda de tensão quando o auscultador do telefone está *on-hook*. Quando o auscultador do telefone está *off-hook*, a carga aplicada à linha faz descer a tensão.

É com base nestes princípios que se pode efetuar o teste das linhas telefónicas, a partir da Central de Comunicações. Para o teste dos telefones e das funcionalidades do sistema, fazem-se, adicionalmente, operações relacionadas com o estabelecimento de chamadas.

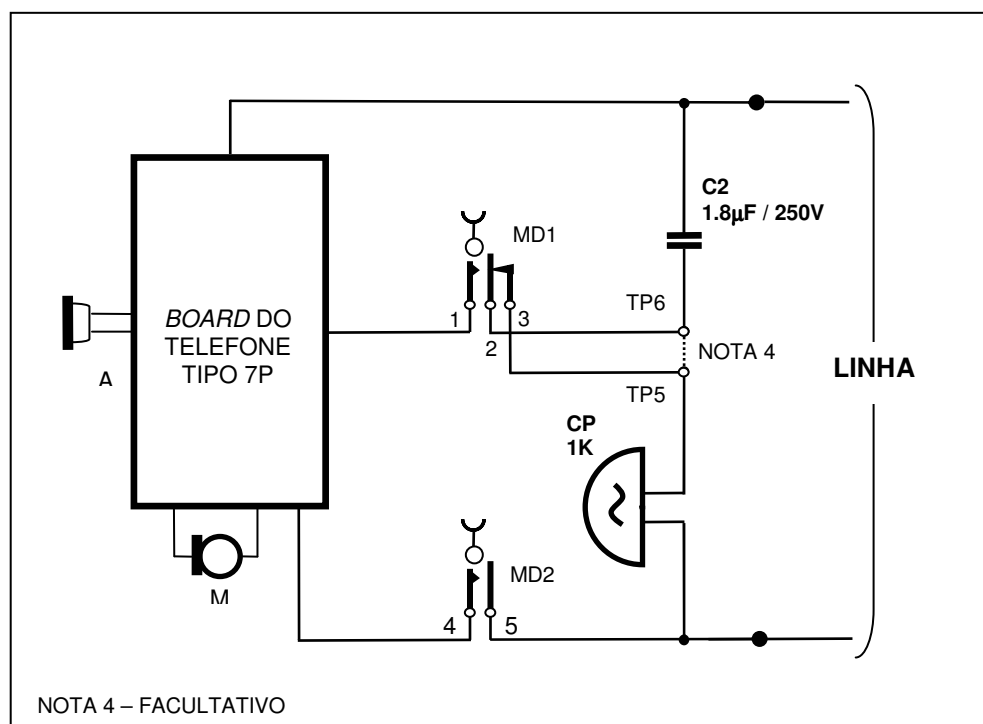


Figura 4.1 – Telefone Tipo 7P – esquema parcial

O sistema de teste aqui descrito foi sobretudo estudado e desenvolvido para teste de linhas analógicas. As linhas digitais RDIS (Rede Digital com Integração de Serviços) / ISDN representavam na altura uma percentagem ínfima do universo das ligações telefónicas. As linhas RDIS são um padrão internacionalmente definido, não havendo especificidades regionais. Por este motivo as atualizações de software respeitavam, sobretudo, a linhas analógicas.

Quando era lançada uma nova versão de software, com novas funcionalidades, o primeiro software que se desenvolvia era para a Alemanha (para a *Deutsche Telekom*), que servia de base (WSA) para adaptações para outros países. De versão para versão eram adicionadas novas funcionalidades (por exemplo, o CLIP (*Calling Line Identification Presentation*), que permite, a quem recebe uma chamada, ver o número de telefone de quem chama). Também, de versão para versão, se adicionava novo hardware, como, por exemplo, novos SLMs (*Subscriber Line Modules*) com maior número de linhas telefónicas, e mais baratos.

4.6. Projeto e implementação

4.6.1. O Equipamento

Um sistema telefónico tem como principais elementos uma Central Telefónica, linhas telefónicas e Telefones.

A Central Telefónica é constituída pelo CP (*Central Processor*) e um ou mais LTGs (*Line Trunk Group*) – Figura 4.2.

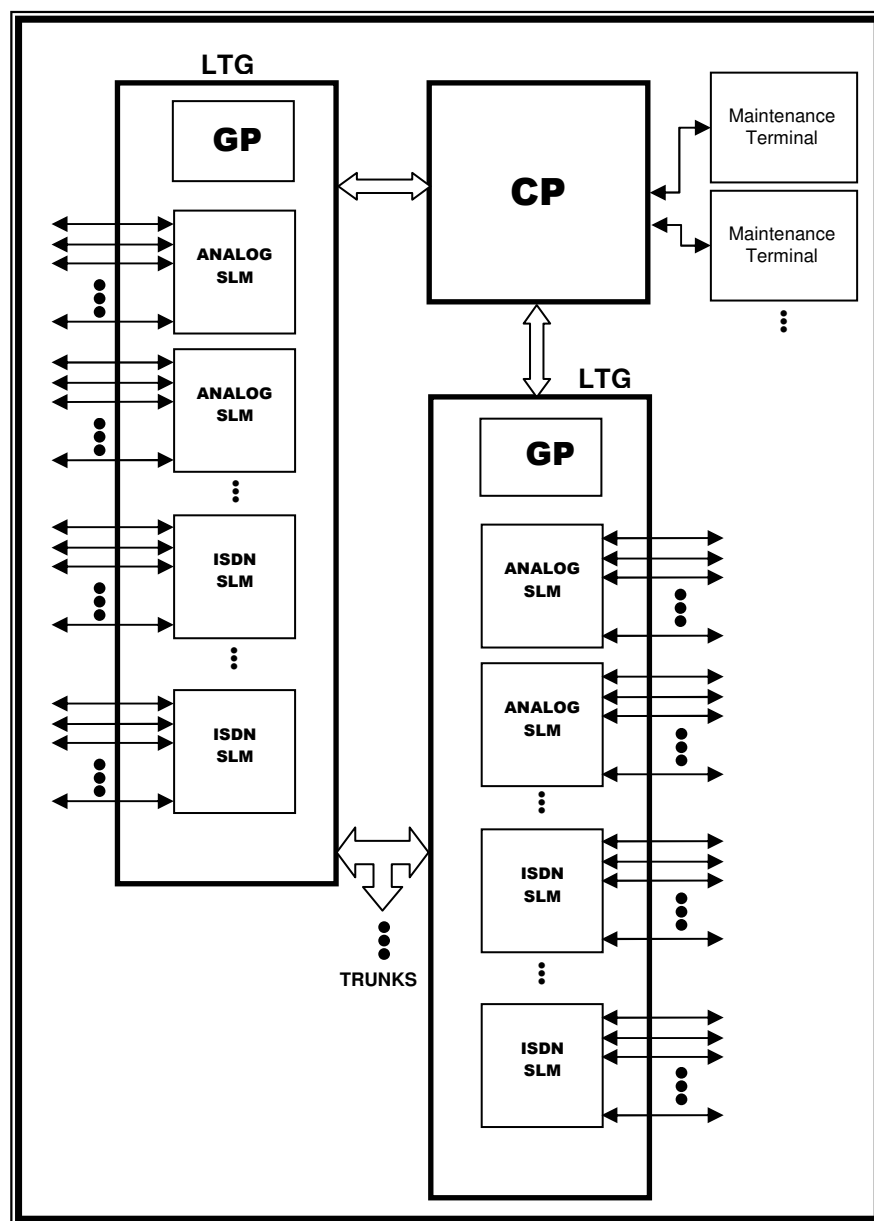


Figura 4.2 – A Central Telefónica – diagrama de blocos

O LTG é o bloco que faz as interfaces com os equipamentos de assinante e outros LTGs. Para chamadas locais, os LTGs ligam a outros LTGs na mesma central.

Os SLMs (*Subscriber Line Modules*) possuem interfaces que são linhas telefónicas, para ligação ao equipamento de assinante. Os primeiros modelos de SLMs possuíam apenas 8 linhas telefónicas. Depois foram aumentando para 16, 32 e 64 linhas. Existem SLMs analógicos e digitais (para linhas RDIS).

As centrais ligam-se entre si através de *trunks*. As *trunks* são canais TDM (*Time Division Multiplexing*) de alta velocidade.

O LTG tem a unidade de processamento GP (*Group Processor*), que estabelece as ligações entre assinantes e faz o controlo e a manutenção das linhas telefónicas. O GP incluía um processador do tipo *Intel '386* e memória RAM.

Em funcionamento normal, o CP tem como principal função o processamento de chamadas. Está ligado a terminais de manutenção.

A linha telefónica é, tipicamente, um *twisted pair*. O *twisted pair* é mais imune ao ruído provocado por interferências eletromagnéticas do que o par direito.

A Figura 4.1 representa o esquema parcial (circuito de entrada) do tradicional telefone *TIPO 7P*, do tipo *Rotary Dial*, usado na rede fixa portuguesa PSTN.

Quando o auscultador do telefone está *on-hook*, o interruptor MD2 está aberto, e o comutador MD1 está na posição entre 2 e 3. Nesta situação, aparece C2 (1.8 μ F) em série com a campainha CP aos terminais da linha telefónica. A carga que o telefone oferece à linha é nula, quando a tensão na linha é DC, o que acontece quando a campainha não está a tocar – a tensão é de 48 V_{DC}.

A campainha toca quando o auscultador do telefone está *on-hook*, e a linha fornece uma tensão alternada (em Portugal é de 20Hz) ao telefone. A tensão aparece aos terminais da campainha, que a faz tocar.

Quando o auscultador do telefone está *off-hook*, o interruptor MD2 está fechado, e o comutador MD1 está na posição entre 1 e 2. Nesta situação o telefone fica preparado para fazer ou atender chamadas.

Os telefones da rede PSTN foram sendo alterados ao longo dos tempos, tornando-se cada vez mais sofisticados, mas os princípios básicos aqui descritos mantiveram-se, o que permite efetuar os testes às linhas e aos telefones sempre da mesma forma e com os mesmos meios.

4.6.2. O Software

O software para o LTG era desenvolvido em *Assembly*. Optou-se, no início (ou seja, há décadas), por esta linguagem porque o software ainda era relativamente simples. Além disso, o *Assembly* permitia um maior domínio de baixo nível do processador. A empresa foi desenvolvendo ferramentas para programação e manutenção fácil e metódica, mantendo no entanto a linguagem *Assembly* para o software do LTG. A mudança para uma linguagem de alto nível nunca aconteceu sobretudo por razões económicas.

Através de comandos MML introduzidos no *Maintenance Terminal* – ver Figura 4.2 – o sistema permite que se efetuem testes às linhas telefónicas e aos telefones.

Os testes que se efetuam estão elencados na Secção 4.3, e o sistema de testes está representado na Figura 4.3.

O TLFI e o TLFB são subsistemas desenvolvidos para estes testes.

O primeiro faz o teste às linhas telefónicas e aos equipamentos de assinantes, sem necessidade de instalar qualquer equipamento adicional.

O TLFB tem o mesmo propósito que o TLFI no que respeita a SLMs analógicos, mas necessita de ter ligado um equipamento específico para teste – o *SULIM*, que substitui o telefone durante os testes. O TLFB é uma aplicação mais antiga que o TLFI, e não testa linhas digitais.

Um dos testes é, por exemplo, testar o sistema de campainha. No caso do TLFI, é mesmo a campainha que toca. No caso do TLFB, é usado o *SULIM*, em substituição do telefone. O *SULIM* tem um conjunto de botões, que se iluminam para cada situação. Quando é enviado sinal para a campainha tocar, o *SULIM* ilumina o botão correspondente.

Para cada teste, há um Comando MML que tem como argumentos os números do LTG, do SLM e da linha.

Para testes específicos (que não têm a ver com chamadas), tais como a medição de tensões, resistências e capacidades, os subsistemas TLFI e TLFB leem os valores através de ADCs incluídas nos SLMs. Em seguida consultam a tabela de *Thresholds*, para verificar se o valor lido está dentro dos limites. Se não estiver, o Comando MML tem como resultado *Falha*.

Os *Thresholds* podiam ser criados e modificados com comandos MML.

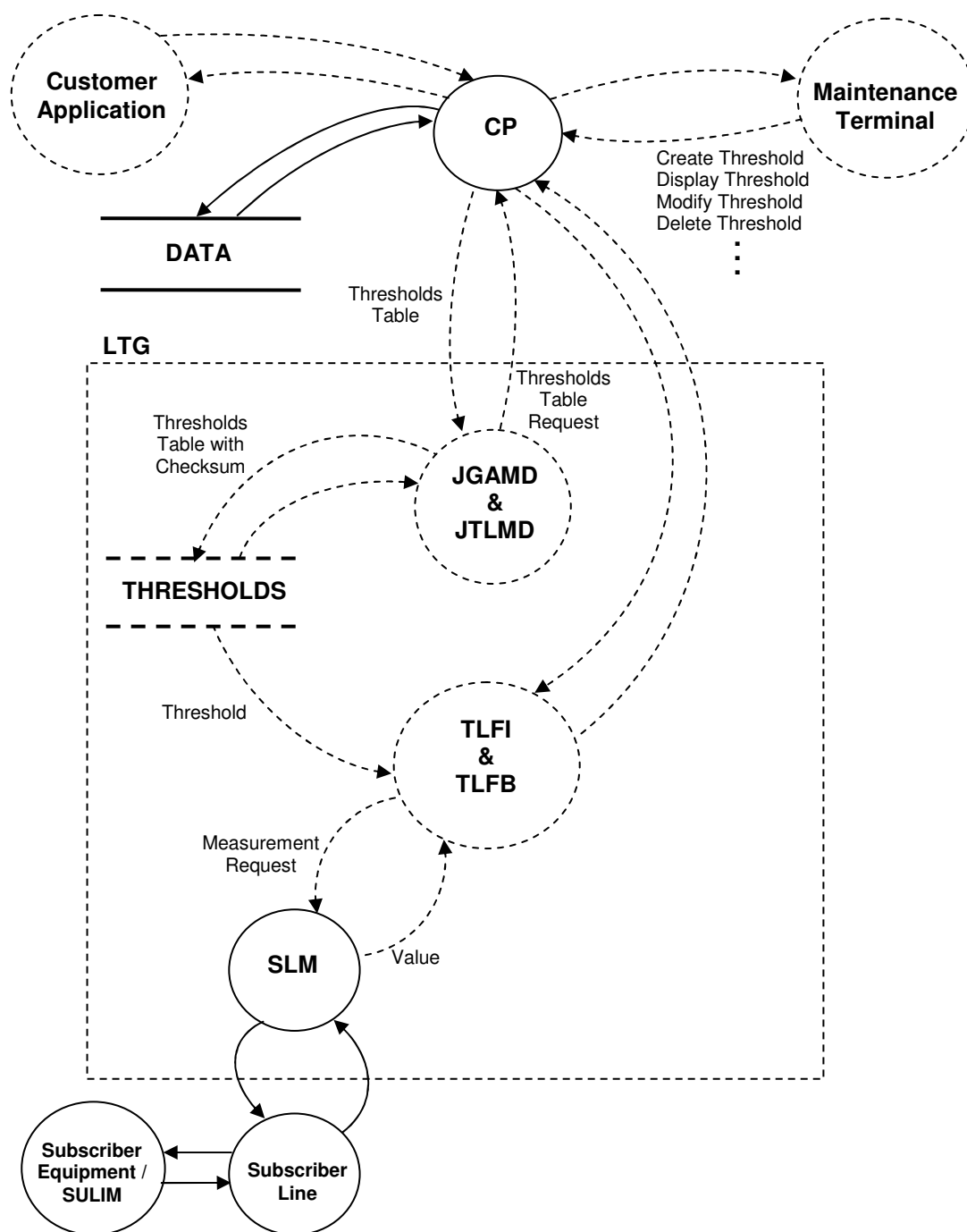


Figura 4.3 – Sistema de teste de linha e telefone

Em linhas digitais RDIS faziam-se, sobretudo, testes no âmbito de chamadas telefónicas. No âmbito do projeto, destinavam-se a teste de novo hardware – novos SLMs.

Mesmo que um subsistema não fosse afetado com nova versão de Software, eram feitos testes de compatibilidade – aquilo que se designava por *Integration Tests*.

4.6.2.1. Subsistemas JGAMD e JTLMD

Os subsistemas JGAMD e JTLMD guardam e garantem a integridade dos *Thresholds* na memória do LTG.

Os *Thresholds* são elementos críticos. Se um *Threshold* estiver errado, podem ocorrer anomalias graves no funcionamento do sistema. Ao serem testadas as linhas telefónicas são consideradas com características fora dos limites (*Thresholds*). Consequentemente, podia-se considerar, incorretamente, que, por exemplo, havia um curto-circuito, e a linha seria desativada.

Os *Thresholds* são enviados ao LTG na sua inicialização, pelo CP. São guardados num bloco de memória de dados do LTG. Como a linguagem de programação do software do LTG era *Assembly*, não existiam limitações ao nível de escrita na memória. Havia algum risco de haver conflitos e sobreposição de dados. Por isso, considerou-se necessário que o bloco de memória onde se guardavam os *Thresholds* fosse permanentemente (i. e. com uma periodicidade de cerca de 1 minuto) auditado e, se algum erro fosse detetado, auto-recuperar.

Os subsistemas JGAMD e JTLMD fazem a *administração* desse bloco de memória, a que se chamou *memória semi-permanente*. Estes subsistemas recebem os *Thresholds*, do CP, e guardam-nos num bloco de memória, onde, no final desse bloco, juntam o respetivo *checksum*.

Com a periodicidade atrás referida, os subsistemas leem a *memória semi-permanente*, e calculam o *checksum*. Se o *checksum* calculado não coincidir com o *checksum* que está no final do bloco, considera-se que a memória está corrompida, e faz-se novo carregamento da memória (dos *Thresholds*), sendo que, para tal, o CP disponibiliza a informação. Tudo isto sem necessidade de intervenção humana – é o chamado *auto-recovery*.

4.6.2.2. A Customer Application

A Central telefónica que a empresa fornecia ao Operador de Telecomunicações tinha todos os meios para um funcionamento pleno. No entanto, ao nível de monitorização ou seja da recolha de dados sobre chamadas telefónicas, e da manutenção automatizada, a empresa não fornecia a aplicação. Era o Operador de Telecomunicações que desenvolvia esse software – *Customer Application*. A Central de Telecomunicações tem uma interface que permite a ligação ao sistema informático do Operador de Telecomunicações. Esta interface fornece informações diversas, tais como quais as chamadas efetuadas, estado das linhas telefónicas e, em geral, o estado da Central. Também permite efetuar algumas configurações, tais como definir números confidenciais. Com base neste recurso a *Customer Application* pode levar a cabo estas operações de forma *user friendly*.

4.7. Resultados

O sistema descrito passava por várias fases de teste (*Offline Test*, *Online Test*, *System Test*), de forma a garantir a satisfação das especificações.

O *Offline Test*, que durava cerca de um mês, era a primeira fase de testes. Os testes eram feitos sem o hardware de destino; faziam-se num computador convencional.

O *Online Test*, que durava cerca de um mês, era feito no hardware de destino do software – na Central Telefónica. O *software developer* executava um, normalmente extenso, conjunto de *test cases* pré-definidas, ao software que desenvolvia.

O *System Test* representava a última fase de testes e era feito por uma equipa alheia ao desenvolvimento de software. O sistema era testado, tanto quanto possível, como num ambiente real, com todas as funcionalidades ativas, e submetido a *stress tests*.

O autor, integrado e apoiado por uma vasta equipa, realizou um trabalho (software) que se revelou fiável, não necessitando de manutenção.

O que o autor achou bastante interessante, neste período da sua vida profissional, foi o conhecimento e seguimento de normas rigorosas estabelecidas na empresa, que impunham a existência de documentação, fases, metas (*milestones*) e métodos pré-definidos; a existência de *ferramentas* de trabalho e controlo da qualidade; a existência, sempre, de um elemento da empresa, que era especialista numa determinada área, ainda que estivesse noutro local do globo, e que podia esclarecer sobre alguma dúvida que existisse. Estes procedimentos asseguravam ao produto padrões elevados de qualidade.

4.8. Bibliografia

- Mayo J. S., *Technical Requirements of the Information Age*
Bell Labs
- Documentação interna da empresa Siemens.

Capítulo 5

Informatização de Laboratórios de Análises Clínicas

5.1. Local e período em que decorreu

Este trabalho decorreu na empresa *Maxdata Informática, Lda.*, de Fevereiro de 1998 a Abril de 1999.

5.2. Introdução

A necessidade da informatização de Laboratórios de Análises Clínicas é indiscutível. Além de oferecerem as vantagens de uma base de dados, os sistemas informáticos de apoio permitem programar os equipamentos que fazem as análises clínicas (os analisadores), indicando quais as

análises a efetuar a cada amostra. Permitem, também, receber dos analisadores, os resultados das análises, proporcionando uma automatização considerável do laboratório diminuindo e facilitando enormemente o trabalho. A informatização pode ser maior ou menor, dependendo se todos ou apenas alguns analisadores são ligados ao sistema informático.

5.3. Objetivos e especificações

Um dos objetivos é poder guardar os dados de pacientes e os resultados de análises de forma segura, fácil e rapidamente acessível. O sistema informático deve ter a capacidade de guardar a informação associada a cada paciente, em quantidades que prevejam o funcionamento do laboratório, durante largos anos. Para cada laboratório é necessário fazer uma estimativa destas necessidades.

O outro objetivo, e que tem a ver diretamente com o trabalho aqui descrito, é tornar mais fácil e rápido o trabalho do técnico de laboratório, evitando a árdua tarefa de, manualmente, programar os analisadores, e transferir os resultados das análises para registo na base de dados. O software de comunicações com os analisadores, integrado no sistema informático do laboratório, oferece estas possibilidades.

Por outro lado, a integridade da informação é um aspeto crítico uma vez que se tratam de dados vitais e confidenciais que condicionam a saúde e a vida das pessoas. Por isso, o software de comunicações deve assegurar que os dados sejam transferidos sem erros.

Para cada analisador existe software de comunicação específico. São *drivers* que permitem ao sistema informático do laboratório comunicar com os analisadores.

Os protocolos de comunicação são muito diversos, mas é possível apresentar algumas especificações genéricas.

Especificações genéricas:

- As comunicações com os analisadores são via RS-232C; tipicamente, têm os *settings*:
 - *Baud rate*: 9600 bps;
 - Paridade: nenhuma;
 - Número de *data bits*: 8;
 - Número de *stop bits*: 1;
- O modo de funcionamento é o *Host Query Mode*, em detrimento do *Batch Mode*. Sempre que não for possível implementar o *Host Query Mode*, o sistema deverá comunicar com os analisadores da forma mais automatizada que cada analisador permitir;

- De forma a garantir a integridade dos *resultados* das análises, o software efetua todos os testes que o protocolo dispuser, para confirmar a validade das tramas, nomeadamente:
 - Processar o campo *checksum* da trama, sempre que o protocolo de comunicações do analisador o possuir, para detetar erros;
 - Confirmar *tokens*, no caso em que a trama os tenha em posições fixas;
 - Confirmar tamanho da trama, no caso de se tratar de trama com comprimento fixo.

5.4. Responsabilidades e competências específicas

Neste projeto o autor implementou os protocolos de comunicação entre os analisadores e o Sistema de Base de Dados. Toda a programação foi realizada em linguagem C.

5.5. Enquadramento teórico

A informatização de um laboratório automatiza consideravelmente o processo de análises clínicas. A utilização de base de dados é indispensável para o bom funcionamento de um laboratório de análises clínicas, permitindo uma fácil organização, facilidade de trabalho, rapidez, segurança e higiene.

A parte da comunicação com os analisadores tem um papel determinante no nível da automatização do processo. Principalmente dela dependerá a eficiência, a produtividade, a fiabilidade e a economia. Por isso, o software de comunicações disponibiliza todas as possibilidades que os analisadores tiverem, para automatizar ao máximo o processo.

Os analisadores possuem interfaces RS-232C que podem ser ligadas a um sistema informático remoto e permitem comunicações a distâncias relativamente razoáveis. Os protocolos de comunicação implementados possibilitam a troca de informação com um sistema informático remoto.

Se o sistema informático do laboratório tiver software preparado para comunicar com o analisador, é possível que o mesmo sistema programe o analisador para que faça as análises pretendidas a uma determinada amostra e receba os resultados dessas análises.

Os protocolos de comunicação existentes são variados. Destaca-se o protocolo *ASTM* (*American Society for Testing and Materials*) que se tornou padrão, e é atualmente bastante usado. Este

protocolo é bastante flexível e oferece elevada segurança das comunicações (as tramas possuem *checksum*). Na Secção 5.6.5.2 fazem-se mais algumas referências sobre este protocolo.

5.6. Projeto e implementação

5.6.1. Introdução

A informatização de Laboratórios de Análises Clínicas implica a instalação de hardware e software.

A empresa para a qual o autor trabalhou instalava o hardware necessário – ver Figura 5.2, à excepção dos analisadores, que eram fornecidos por companhias especializadas.

Ao nível do software há uma base de dados e software para comunicação com os analisadores.

O software mínimo que se instalava incluía apenas a base de dados. O software de comunicação com os analisadores era fornecido à parte.

O Sistema de Análises Clínicas está representado, de forma estruturada na Figura 5.1.

Um acontecimento de análises clínicas começa com uma requisição médica.

De seguida, o paciente desloca-se ao posto de colheita para recolha de amostras de sangue, urina, etc. Abre-se uma ficha na base de dados, com dados pessoais do paciente e a lista de análises requerida pelo médico. Terminada esta operação, o software cria (faz o *append*) um registo na *Lista de Trabalho* para indicar ao analisador quais as análises a efetuar à amostra colhida. Esta informação é enviada ao analisador. As amostras do paciente são colocadas no analisador. De seguida, o analisador efetua as análises de acordo com a *Lista de Trabalho*.

Terminadas as análises, o analisador envia ao sistema informático do laboratório os resultados das análises. Os resultados são guardados temporariamente no armazém *Resultados*.

O acontecimento termina com a validação dos resultados e a extração do boletim que contém a identificação do paciente e os resultados das análises. É o *output* do sistema. No Apêndice B , apresenta-se um exemplo de boletim.

A validação dos resultados é uma operação efetuada pelo médico do laboratório. O software lê a informação contida no armazém *Resultados* e exibe-a para que o médico verifique a sua conformidade. Ao validarem-se os resultados, estes são guardados, de forma permanente, na base de dados e os ficheiros *Lista de Trabalho* e *Resultados* (cada analisador tem um ficheiro para *Lista de Trabalho* e outro para *Resultados*) são esvaziados.

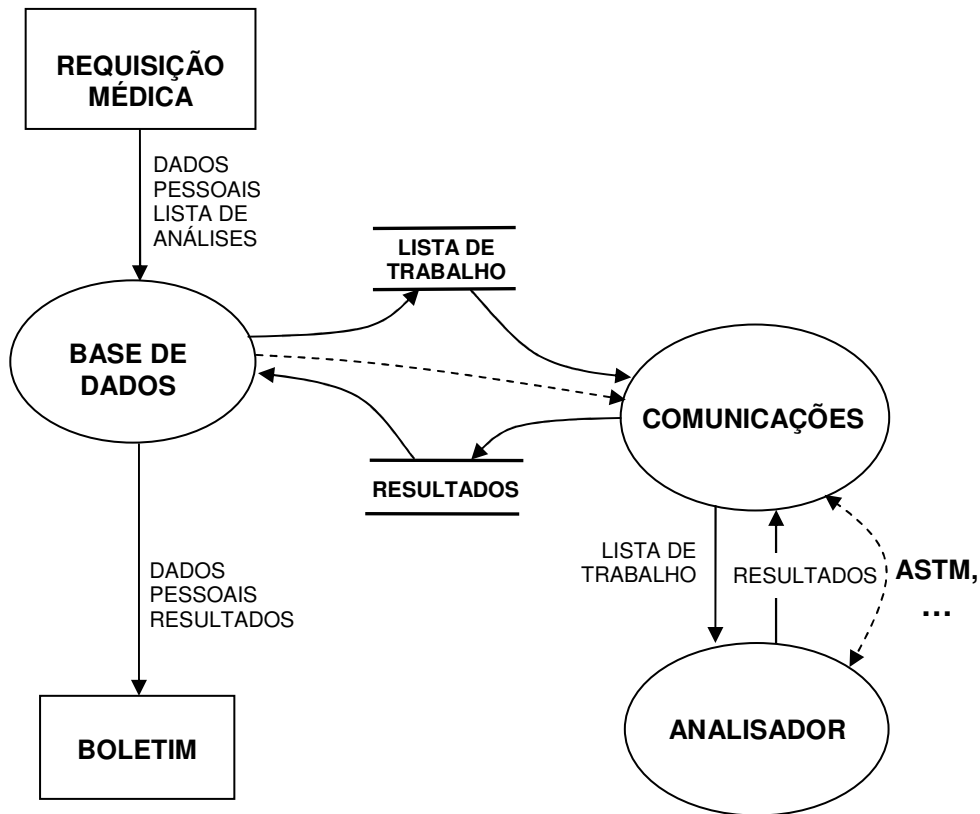


Figura 5.1 – O Sistema Informático de Análises Clínicas

5.6.2. O Equipamento

A Figura 5.2 representa o equipamento típico instalado na informatização de laboratórios de análises clínicas.

Na *Recepção* é atendido o paciente ou o seu representante, recebida a requisição de análises médicas, e aberta uma ficha no sistema informático onde constam os dados pessoais do paciente e a lista de análises requerida pelo médico.

No *Posto de colheita* são extraídas amostras do paciente, para análise, de acordo com receita médica. As amostras são colocadas num tubo de laboratório, ao qual se cola uma etiqueta com *código de barras* para identificação.

No *Centro informático* está o computador.

Um equipamento periférico – *RS-232C Batch*, externo ao computador, que disponibiliza os portos RS-232C necessários, é ligado ao PC através da interface *Ethernet*. Este equipamento liga o computador aos analisadores, aos terminais de utilizador e algumas impressoras, através de portos RS-232C.

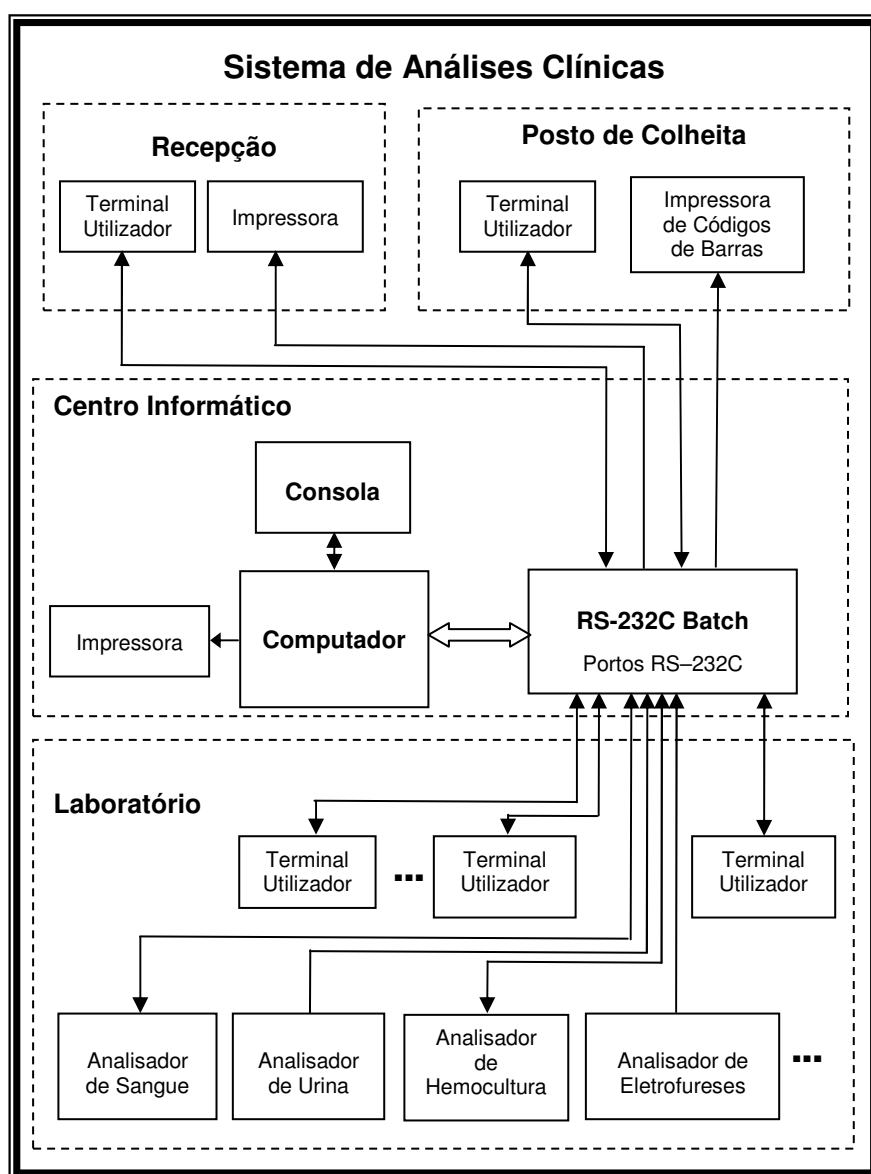


Figura 5.2 – Sistema de análises clínicas – o equipamento

5.6.3. O Software do Sistema Operativo

O sistema operativo escolhido para o sistema informático foi o *UNIX da SCO (Santa Cruz Operation)*. Este sistema tem uma boa gestão de processamento multi-tarefa, sendo que os processos não ficam tempo excessivo sem terem o seu *timeslice*.

O equipamento *RS-232C Batch* vem acompanhado de *drivers* para o UNIX. Uma vez instalado este equipamento, cada um dos seus portos RS-232C apresenta-se como se fosse um porto interno do computador (tipo COM1, COM2, ...).

Um porto RS-232C pode ser configurado para fornecer uma sessão *UNIX*, para terminal de utilizador, ou para comunicações genéricas RS-232C (com os analisadores, etc). Os comandos do *UNIX enable* e *disable* permitem a configuração pretendida. Para configurar o porto *tty1a*, para ser ligado a terminal de utilizador, executa-se o comando *enable tty1a*. O comando *disable tty1a* faz o porto regressar ao modo de comunicações genéricas RS-232C. À partida, os portos estão no estado *disable*, pelo que basta fazer o *enable* daqueles que estão ligados aos terminais de utilizador.

5.6.4. A Base de Dados

A base de dados foi implementada em *COBOL* (*COmmon Business Oriented Language*). O autor não integrava a equipa que trabalhava na base de dados.

A base de dados tinha a seguinte interação com o Software de Comunicações com os analisadores:

- Fornecia a *Lista de Trabalho* – ver Figura 5.1 – em ficheiro texto;
- Recebia, do Software de Comunicações com os analisadores, os *resultados* das análises – ver Figura 5.1 – em ficheiro texto;
- Fazia a gestão dos ficheiros da *Lista de Trabalho* e *Resultados*; quando os *resultados* eram validados, esses ficheiros eram esvaziados;
- Ativava e desativava as comunicações com os analisadores.

5.6.5. Software de Comunicações com os analisadores

Neste projeto o autor foi responsável pela implementação desta parte do software. O software foi escrito em linguagem *C* para correr sobre o sistema operativo *UNIX*.

Nas comunicações com os analisadores há que evitar a perda de informação durante a recepção das tramas. Este é o aspeto mais crítico mas consegue-se facilmente, desde que o processo que recebe as tramas não efetue outros processamentos durante a recepção destas especialmente acessos ao espaço I/O, como operações *printf*, *fprintf*, que são tipicamente lentas. Só depois de

recebida a trama é que se faz o seu processamento e se acede ao espaço I/O (disco, ecrã). Durante a recepção da trama, usa-se apenas a memória RAM para guardar os dados.

A Figura 5.3 representa o fluxograma típico do software de comunicações entre o sistema informático do laboratório e o analisador.

Quando é recebida uma trama vinda de analisador, verifica-se se é um pedido de indicação de quais as análises a efectuar a determinada amostra (se é uma trama do tipo *Query*). Se for, responde-se indicando quais as análises a efectuar. Se não for, verifica-se se contém resultados. Em caso afirmativo, guardam-se em ficheiro.

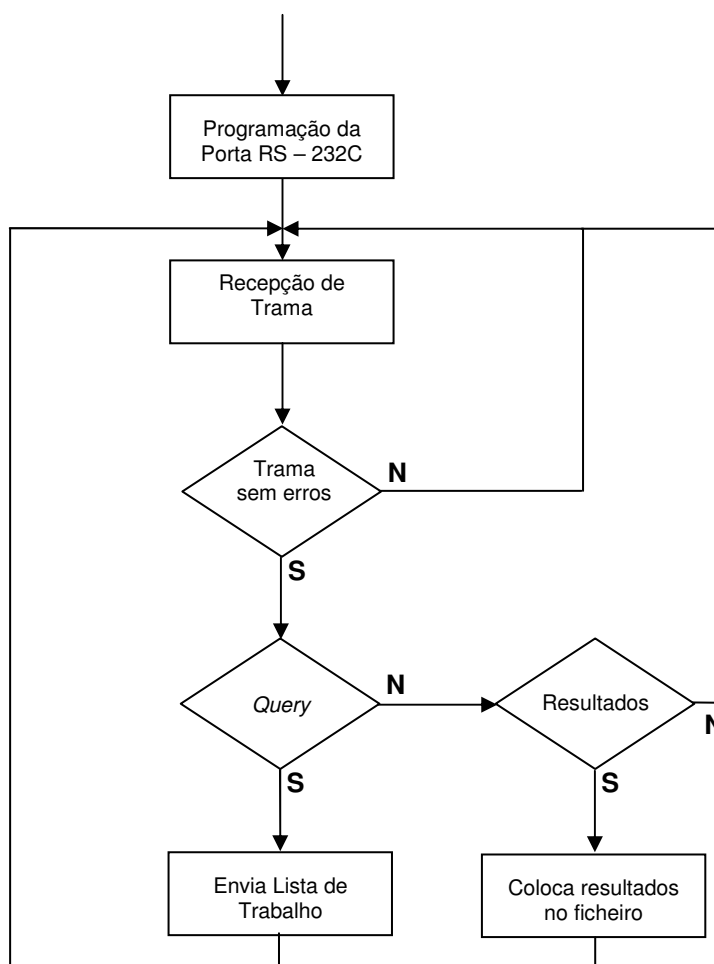


Figura 5.3 – Fluxograma do software de comunicações

Neste fluxograma não estão representadas as comunicações de controlo.

Os analisadores possuíam protocolos de comunicação muito diversos, especialmente os analisadores mais antigos. Nos analisadores mais recentes, estava a usar-se, cada vez mais, o protocolo padrão – o *ASTM* – descrito sucintamente na secção 5.6.5.2.

5.6.5.1. Protocolos proprietários

Nos protocolos mais simples (normalmente proprietários), a comunicação é unidirecional: do analisador para o sistema informático do laboratório. O analisador é programado manualmente. O analisador efetua as análises e envia os resultados ao sistema informático do laboratório sem receber qualquer *confirmação*. É o técnico de laboratório que tem de confirmar se a informação foi recebida. Neste tipo de protocolos, geralmente o software apenas faz o controlo da informação recebida através da confirmação do tamanho da trama.

5.6.5.2. Protocolo ASTM

O protocolo cada vez mais adoptado na transferência de informação entre equipamentos de análises clínicas (analisadores) e os sistemas informáticos dos laboratórios, e que acabou por se tornar padrão, foi o *ASTM* (*American Society for Testing and Materials*). Este protocolo prevê comunicação bilateral entre o analisador e o sistema informático do laboratório. São transferidas tramas de controlo e informação. As tramas de informação possuem *checksum*.

5.6.5.3. Funcionamento Batch Mode

Numa das formas de utilização, o sistema informático do laboratório envia ao analisador, em resultado de comando do técnico de análises clínicas, uma tabela (*batch*) de análises a efetuar a um conjunto de amostras (envia a *Lista de Trabalho*). De seguida, o analisador efetua as análises, de acordo com a *Lista de Trabalho*. Terminadas as análises, o analisador envia os resultados para o sistema informático do laboratório, após comando do técnico de análises. A receção das tramas normalmente tem uma mensagem de confirmação.

5.6.5.4. Funcionamento Host Query Mode

A utilização mais automatizada do Sistema é designada por *Host Query Mode*.

Ao ser colocada a amostra de uma colheita no analisador, este lê um código de barras colado no tubo da amostra, que a identifica. De seguida, o analisador (*Host*) inquirir (*Query*) o sistema informático do laboratório para que o informe sobre quais as análises a efetuar àquela amostra. Em resposta, o sistema informático envia a informação, relativa àquela amostra, contida na *Lista de Trabalho*, ao analisador. Este efetua as análises de acordo com as instruções que recebe da base de dados, e, terminadas as análises, envia os resultados.

O funcionamento de um sistema deste tipo é bastante automático, não requerendo trabalho do técnico de análises clínicas, além do que é necessário para colocar as amostras no analisador.

O *Host Query Mode* estava predominantemente implementado em analisadores cujo protocolo de comunicações era o *ASTM*. No entanto, havia alguns com protocolos proprietários que também possibilitavam este modo de funcionamento.

5.6.6. Outros dados

Entre os laboratórios informatizados com o sistema desenvolvido constam-se:

Laboratórios públicos:

- Hospital de Santa Maria
- Hospital Fernando Fonseca (Hospital Amadora – Sintra)
- Hospital Pulido Valente
- Maternidade Alfredo da Costa
- Hospital Pedro Hispano (Matosinhos)
- Hospital de Viana do Castelo

Laboratórios privados

- Fernando Teixeira (Lisboa)
- Reis Valle (Lisboa)

5.7. Resultados

Ao nível da utilização do sistema informático, os resultados traduziram-se numa maior organização e rapidez, trabalho mais simples e menos maçador, e maior fiabilidade dos registos.

Este projeto permitiu ao autor adquirir muita experiência a programar em linguagem *C*. Entre outras coisas pôde constatar que, na área do software de comunicações, se podem construir programas com aspeto desde o nível mais baixo até níveis elevados de abstração

Enquanto trabalhou nesta empresa, o autor teve um curso de programação em *Delphi*, prestado pelo representante em Portugal da *Borland*. O principal objetivo era passar do sistema informático com interface alfanumérica para interface gráfica.

5.8. Bibliografia

- Abbott, *Hosting Elecsys 2010*
- Menarini Diagnostics, *Super Auction 4250*
- Colclasser R. A. & Diehl-Nagle, *Materials and Devices*
McGRAW-HILL

Capítulo 6

Automatismos

6.1. Local e período em que decorreu

Em finais de 1997 e inícios de 1998, o autor trabalhou na empresa *DASEL – Desenvolvimento de Aplicações e Sistemas de Electrónica, Lda.*, em Oeiras.

6.2. Introdução

A empresa referida instalava automatismos, que separava em dois grupos:

- Equipamentos de desenfumagem
- Automatismos genéricos, nomeadamente Grades, Barreiras e Portas Automáticas

A desenfumagem é destinada a situações de emergência, em que a atmosfera no interior de um edifício está contaminada, ou está em risco iminente de contaminação, especialmente em situações de incêndio. Nestas situações, é acionado (normalmente por bombeiros) o sistema de desenfumagem. Ao ser acionado, o sistema abre as janelas, evitando a intoxicação e tornando mais lenta a propagação do fogo.

Quando a atmosfera no interior dos edifícios fica contaminada é fundamental que se abram as janelas para que o ar continue respirável e se protejam as pessoas.

Os sistemas de desenfumagem são, sobretudo, instalados em edifícios abertos ao público.

A empresa tinha o seu interesse focado na área da desenfumagem e era a única no país que comercializava equipamentos e prestava serviços nesta área.

6.3. Objetivos e especificações

A palavra automatismo sugere, em termos genéricos, o que se pretende. Ou seja, refere-se à substituição de tarefas manuais por máquinas. Idealmente o funcionamento destas máquinas ou dispositivos deve ter o mínimo de intervenção humana. Além disso pretende-se que pessoas e bens estejam protegidos, e, se acontecer algum incidente, se minimizarem os danos.

Na DASEL, o trabalho consistia em fazer projetos de instalações de automatismos, e instalá-los de modo a satisfazer as necessidades dos clientes.

6.4. Responsabilidades e competências específicas

Nesta empresa, as funções desempenhadas pelo autor foram, sobretudo, de elaboração de projetos de instalações de automatismos. Configurava esses equipamentos e orientava a equipa que os instalava. Ocasionalmente contactava fornecedores e clientes.

6.5. Enquadramento teórico

Um automatismo tem por base um motor elétrico. O motor, ao ser acionado, provoca um movimento de rotação do seu eixo que é transmitido a outras partes móveis de modo a reduzir o trabalho. A barreira automática é um exemplo em que o efeito final é de rotação. A rotação do eixo da barreira fá-la levantar e baixar (abrir e fechar). O atuador elétrico usado na desenfumagem, efetua movimentos retilíneos do seu cursor, que fazem abrir e fechar as janelas e arejar o edifício.

6.6. Projeto e implementação

6.6.1. Introdução

Na área da desenfumagem, a empresa instalava dois tipos de equipamentos: elétricos e pneumáticos. Os elétricos eram os mais frequentes.

Na área dos automatismos genéricos, cada situação era analisada, e era executado um projeto que satisfizesse as necessidades.

6.6.2. O sistema de desenfumagem elétrico

A Figura 6.1 mostra o equipamento típico que se instala num sistema de desenfumagem elétrico. O sistema é constituído por uma central de controlo, um ou mais comandos para acionamento e atuadores elétricos que se instalam nas janelas, que as abrem e fecham.

O acionamento do automatismo é manual – nos *comandos*. Os cabos de ligação entre os atuadores e a central, e a central e os comandos são anti-fogo.

Numa situação de emergência há corte de energia. Se o corte não for espontâneo (por sobrecarga ou curto-circuito), a energia é desligada manualmente por razões de segurança. Por isso, a central possui baterias de reserva – UPS (*Uninterruptible Power Supply*).

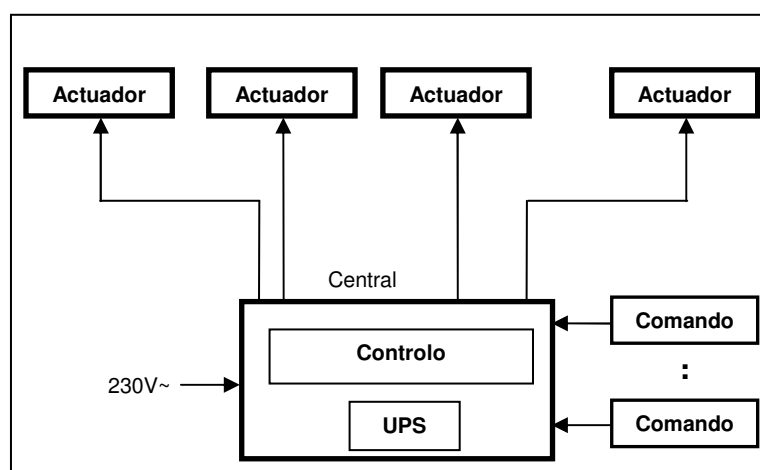


Figura 6.1 – Sistema de desenfumagem elétrico

Neste tipo de desenfumagem o dimensionamento passa pela selecção do atuador adequado para cada janela – curso, potência, mecânica de funcionamento. A Figura 6.2 representa um exemplo de instalação de atuador em janela. O deslocamento do cursor do atuador só é possível quando se faz rodar o respetivo motor, ou seja, quando o motor é alimentado, o que impede que a janela seja aberta intempestivamente.

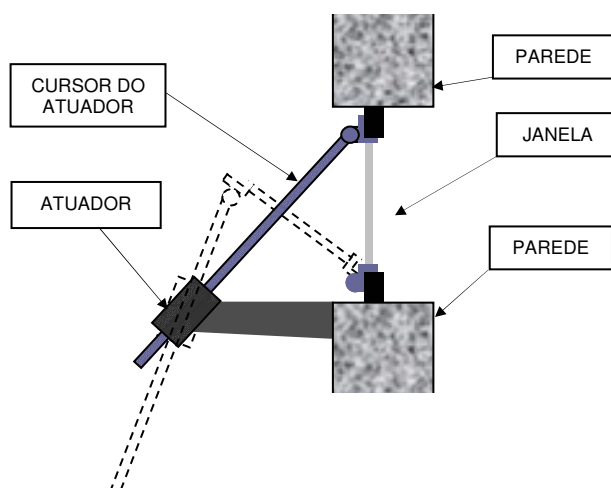


Figura 6.2 – Atuador instalado em janela

A Central é dimensionada em função da carga que depende do número de atuadores e da potência total.

6.7. Alguns Clientes

Durante a estadia do autor na DASEL, foram instalados sistemas de desenfumagem, entre outros, nos seguintes locais:

- Hotel Mundial – Lisboa
- Hotel D. Pedro – Lisboa
- Hotel no Estoril
- Alguns edifícios da *EXPO98*
- Casino da Figueira da Foz
- Norte Shopping – Vila Nova de Gaia

6.8. Resultados

Na área da desenfumagem, com as instalações feitas, protegeram-se os edifícios contra tragédias e danos elevados. Com os sistemas descritos garante-se que a atmosfera no interior dos edifícios continua respirável em caso de incêndio ou contaminação, e que os danos sejam mínimos.

Este trabalho permitiu ao autor conhecer algumas das muitas formas de como o motor elétrico, quando controlado por sistemas eletrónicos adequados, pode ser usado para automação. Com esta experiência o autor adquiriu experiência profissional na área de projetos de automação e controlo.

6.9. Bibliografia

- Avalone & Baumeister, *Standard Handbook for Mechanical Engineers*
McGRAW-HILL
- Beeman, *Industrial Power Systems Handbook*
McGRAW-HILL

- Documentação de automatismos – (PE, ...).

Capítulo 7

Docência no Ensino Secundário

No ano letivo de 1996/1997, o autor foi professor na Escola Secundária de Ourém, Concelho de onde é natural. Leccionou, ao 10º ano de Escolaridade, *Práticas Oficiais e Laboratoriais (POL)*. Do programa da disciplina constavam:

- Higiene e Segurança no Local de Trabalho
- Instalações Elétricas
- Instalações Telefónicas

O autor foi Delegado de Grupo de *POL*.

No final do ano letivo, o autor substituiu o colega Professor de Sistemas Digitais, do 12º ano, que, por motivos de doença, se ausentou. Ensinou o funcionamento de microprocessadores. Fez a correcção do exame de Sistemas Digitais, de aferição à Universidade, que deixou disponível aos alunos.

Terminadas as aulas, trabalhou no lançamento das classificações para acesso à Universidade.

No Ano Letivo seguinte, o autor foi colocado em Portalegre a leccionar Matemática, ao 3º Ciclo do Ensino Básico. Ainda no primeiro trimestre, o autor abandonou o Ensino, e foi trabalhar na empresa de automatismos, referida no Capítulo 6.

Apêndice A – MIB-II (excertos)

```
RFC1213-MIB DEFINITIONS ::= BEGIN

IMPORTS
    mgmt, NetworkAddress, IPAddress, Counter, Gauge,
        TimeTicks
        FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC-1212;

-- This MIB module uses the extended OBJECT-TYPE macro as
-- defined in [14];

-- MIB-II (same prefix as MIB-I)

mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }

-- textual conventions

DisplayString ::=
    OCTET STRING
-- This data type is used to model textual information taken
-- from the NVT ASCII character set. By convention, objects
-- with this syntax are declared as having
--
--     SIZE (0..255)

PhysAddress ::=
    OCTET STRING
-- This data type is used to model media addresses. For many
-- types of media, this will be in a binary representation.
-- For example, an ethernet address would be represented as
-- a string of 6 octets.

-- groups in MIB-II

system      OBJECT IDENTIFIER ::= { mib-2 1 }

interfaces  OBJECT IDENTIFIER ::= { mib-2 2 }

at          OBJECT IDENTIFIER ::= { mib-2 3 }

ip          OBJECT IDENTIFIER ::= { mib-2 4 }

icmp        OBJECT IDENTIFIER ::= { mib-2 5 }

tcp         OBJECT IDENTIFIER ::= { mib-2 6 }

udp         OBJECT IDENTIFIER ::= { mib-2 7 }

egp         OBJECT IDENTIFIER ::= { mib-2 8 }

-- historical (some say hysterical)
-- cmot      OBJECT IDENTIFIER ::= { mib-2 9 }

transmission OBJECT IDENTIFIER ::= { mib-2 10 }

snmp        OBJECT IDENTIFIER ::= { mib-2 11 }

-- the System group

-- Implementation of the System group is mandatory for all
-- systems. If an agent is not configured to have a value
-- for any of these variables, a string of length 0 is
-- returned.

sysDescr OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "A textual description of the entity. This value
        should include the full name and version
        identification of the system's hardware type,
        software operating-system, and networking
        software. It is mandatory that this only contain
        printable ASCII characters."
    ::= { system 1 }

-- the Interfaces group

-- Implementation of the Interfaces group is mandatory for
-- all systems.

ifNumber OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of network interfaces (regardless of
        their current state) present on this system."
    ::= { interfaces 1 }

-- the Interfaces table

-- The Interfaces table contains information on the entity's
-- interfaces. Each interface is thought of as being
-- attached to a 'subnetwork'. Note that this term should
-- not be confused with 'subnet' which refers to an
-- addressing partitioning scheme used in the internet suite
-- of protocols.

ifTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IfEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A list of interface entries. The number of
        entries is given by the value of ifNumber."
    ::= { interfaces 2 }

ifEntry OBJECT-TYPE
    SYNTAX  IfEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "An interface entry containing objects at the
        subnetwork layer and below for a particular
        interface."
    INDEX   { ifIndex }
    ::= { ifTable 1 }

IfEntry ::=
    SEQUENCE {
        ifIndex
            INTEGER,
        ifDescr
            DisplayString,
        ifType
            INTEGER,
        ifMtu
            INTEGER,
        ifSpeed
            Gauge,
        ifPhysAddress
            PhysAddress,
        ifAdminStatus
            INTEGER,
        ifOperStatus
            INTEGER,
        ifLastChange
            TimeTicks,
```

```

ifInOctets          ultra(29),          -- ULTRA technologies
    Counter,        ds3(30),           -- T-3
ifInUcastPkts       sip(31),           -- SMDS
    Counter,        frame-relay(32)
ifInNUcastPkts      }
    Counter,        ACCESS read-only
ifInDiscards        STATUS mandatory
    Counter,        DESCRIPTION
ifInErrors           "The type of interface, distinguished according to
    Counter,        the physical/link protocol(s) immediately 'below'
ifInUnknownProtos   the network layer in the protocol stack."
    Counter,        ::= { ifEntry 3 }
ifOutOctets         (... )
    Counter,
ifOutUcastPkts      }
    Counter,
ifOutNUcastPkts     }
    Counter,        -- the IP group
ifOutDiscards       -- Implementation of the IP group is mandatory for all
    Counter,        -- systems.
ifOutErrors         ipForwarding OBJECT-TYPE
    Counter,        SYNTAX INTEGER {
ifOutQLen           forwarding(1),      -- acting as a gateway
    Gauge,          not-forwarding(2) -- NOT acting as a gateway
ifSpecific          }
    OBJECT IDENTIFIER
}
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "The indication of whether this entity is acting
    as an IP gateway in respect to the forwarding of
    datagrams received by, but not addressed to, this
    entity. IP gateways forward datagrams. IP hosts
    do not (except those source-routed via the host).

    Note that for some managed nodes, this object may
    take on only a subset of the values possible.
    Accordingly, it is appropriate for an agent to
    return a 'badValue' response if a management
    station attempts to change this object to an
    inappropriate value."
::= { ifEntry 1 }

ifIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "A unique value for each interface. Its value
    ranges between 1 and the value of ifNumber. The
    value for each interface must remain constant at
    least from one re-initialization of the entity's
    network management system to the next re-
    initialization."
::= { ifEntry 1 }

ifDescr OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "A textual string containing information about the
    interface. This string should include the name of
    the manufacturer, the product name and the version
    of the hardware interface."
::= { ifEntry 2 }

ifType OBJECT-TYPE
SYNTAX INTEGER {
    other(1),          -- none of the following
    regular1822(2),
    hdh1822(3),
    ddn-x25(4),
    rfc877-x25(5),
    ethernet-csmacd(6),
    iso88023-csmacd(7),
    iso88024-tokenBus(8),
    iso88025-tokenRing(9),
    iso88026-man(10),
    starLan(11),
    proteon-10Mbit(12),
    proteon-80Mbit(13),
    hyperchannel(14),
    fddi(15),

    lapb(16),
    sdlc(17),
    ds1(18),          -- T-1
    el(19),          -- european equiv. of T-1
    basicISDN(20),
    primaryISDN(21), -- proprietary serial
    propPointToPointSerial(22),
    ppp(23),
    softwareLoopback(24),
    eon(25),          -- CLNP over IP [11]
    ethernet-3Mbit(26),
    nsip(27),         -- XNS over IP
    slip(28),         -- generic SLIP
}

ipDefaultTTL OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "The default value inserted into the Time-To-Live
    field of the IP header of datagrams originated at
    this entity, whenever a TTL value is not supplied
    by the transport layer protocol."
::= { ip 2 }

ipAddrTable OBJECT-TYPE
SYNTAX SEQUENCE OF IpAddrEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "The table of addressing information relevant to
    this entity's IP addresses."
::= { ip 20 }

ipAddrEntry OBJECT-TYPE
SYNTAX IpAddrEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "The addressing information for one of this
    entity's IP addresses."
INDEX { ipAdEntAddr }
::= { ipAddrTable 1 }

IpAddrEntry ::=

```

```

SEQUENCE {
    ipAdEntAddr
        IpAddress,
    ipAdEntIfIndex
        INTEGER,
    ipAdEntNetMask
        IpAddress,
    ipAdEntBcastAddr
        INTEGER,
    ipAdEntReasmMaxSize
        INTEGER (0..65535)
}

ipAdEntAddr OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The IP address to which this entry's addressing
        information pertains."
    ::= { ipAddrEntry 1 }

(...)

-- the IP routing table

-- The IP routing table contains an entry for each route
-- presently known to this entity.

ipRouteTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IpRouteEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This entity's IP Routing table."
    ::= { ip 21 }

ipRouteEntry OBJECT-TYPE
    SYNTAX  IpRouteEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A route to a particular destination."
    INDEX   { ipRouteDest }
    ::= { ipRouteTable 1 }

IpRouteEntry ::=
    SEQUENCE {
        ipRouteDest
            IpAddress,
        ipRouteIfIndex
            INTEGER,
        ipRouteMetric1
            INTEGER,
        ipRouteMetric2
            INTEGER,
        ipRouteMetric3
            INTEGER,
        ipRouteMetric4
            INTEGER,
        ipRouteNextHop
            IpAddress,
        ipRouteType
            INTEGER,
        ipRouteProto
            INTEGER,
        ipRouteAge
            INTEGER,
        ipRouteMask
            IpAddress,
        ipRouteMetric5
            INTEGER,
        ipRouteInfo
            OBJECT IDENTIFIER
    }

(...)

-- the IP Address Translation table

-- The IP address translation table contain the IpAddress to
-- 'physical' address equivalences. Some interfaces do not
-- use translation tables for determining address
-- equivalences (e.g., DDN-X.25 has an algorithmic method);
-- if all interfaces are of this type, then the Address
-- Translation table is empty, i.e., has zero entries.

ipNetToMediaTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IpNetToMediaEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "The IP Address Translation table used for mapping
        from IP addresses to physical addresses."
    ::= { ip 22 }

(...)

-- additional IP objects

ipRoutingDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of routing entries which were chosen
        to be discarded even though they are valid. One
        possible reason for discarding such an entry could
        be to free-up buffer space for other routing
        entries."
    ::= { ip 23 }

-- the ICMP group

-- Implementation of the ICMP group is mandatory for all
-- systems.

icmpInMsgs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of ICMP messages which the
        entity received. Note that this counter includes
        all those counted by icmpInErrors."
    ::= { icmp 1 }

(...)

-- the TCP group

-- Implementation of the TCP group is mandatory for all
-- systems that implement the TCP.

-- Note that instances of object types that represent
-- information about a particular TCP connection are
-- transient; they persist only as long as the connection
-- is in question.

tcpRtoAlgorithm OBJECT-TYPE
    SYNTAX  INTEGER {
        other(1),    -- none of the following
                    constant(2), -- a constant rto
                    rsre(3),    -- MIL-STD-1778, Appendix B
                    vanj(4)     -- Van Jacobson's algorithm [10]
    }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The algorithm used to determine the timeout value
        used for retransmitting unacknowledged octets."
    ::= { tcp 1 }

(...)

-- the TCP Connection table

-- The TCP connection table contains information about this
-- entity's existing TCP connections.

tcpConnTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF TcpConnEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A table containing TCP connection-specific

```

```
information."
::= { tcp 13 }

-- additional TCP objects

tcpInErrs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of segments received in error
        (e.g., bad TCP checksums)."
    ::= { tcp 14 }

tcpOutRsts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of TCP segments sent containing the
        RST flag."
    ::= { tcp 15 }

-- the UDP group

-- Implementation of the UDP group is mandatory for all
-- systems which implement the UDP.

udpInDatagrams OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of UDP datagrams delivered to
        UDP users."
    ::= { udp 1 }

udpNoPorts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of received UDP datagrams for
        which there was no application at the destination
        port."
    ::= { udp 2 }

udpInErrors OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of received UDP datagrams that could
        not be delivered for reasons other than the lack
        of an application at the destination port."
    ::= { udp 3 }

udpOutDatagrams OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of UDP datagrams sent from this
        entity."
    ::= { udp 4 }

-- the UDP Listener table

-- The UDP listener table contains information about this
-- entity's UDP end-points on which a local application is
-- currently accepting datagrams.

udpTable OBJECT-TYPE
    SYNTAX SEQUENCE OF UdpEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A table containing UDP listener information."
    ::= { udp 5 }

-- the SNMP group

-- Implementation of the SNMP group is mandatory for all
-- systems which support an SNMP protocol entity. Some of
-- the objects defined below will be zero-valued in those
-- SNMP implementations that are optimized to support only
-- those functions specific to either a management agent or
-- a management station. In particular, it should be
-- observed that the objects below refer to an SNMP entity,
-- and there may be several SNMP entities residing on a
-- managed node (e.g., if the node is hosting acting as
-- a management station).

snmpInPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of Messages delivered to the
        SNMP entity from the transport service."
    ::= { snmp 1 }

snmpOutPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP Messages which were
        passed from the SNMP protocol entity to the
        transport service."
    ::= { snmp 2 }

snmpInBadVersions OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP Messages which were
        delivered to the SNMP protocol entity and were for
        an unsupported SNMP version."
    ::= { snmp 3 }

snmpInBadCommunityNames OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of SNMP Messages delivered to
        the SNMP protocol entity which used a SNMP
        community name not known to said entity."
    ::= { snmp 4 }
```


Apêndice B – Boletim Clínico

Pag.1

Exmo.Senhor

Analises requisitadas por:

Data Colheita:
Data Saída :

NT:249 NC:71745 NMec:0503100
Serviço: Piso: Cama:

ANALISES	RESULTADOS / UNIDADES	VALORES REFERENCIA
----------	-----------------------	--------------------

HEMATOLOGIA

HEMOGRAMA

ERITROGRAMA:

Eritrócitos.....	5.67	x10 ⁶ /uL
Hemoglobina.....	16.2	g/dL
Hematócrito.....	50.2	%
Vol. Glob. Médio...	88.5	fL
Hemog. Glob. Média.	28.6	pg
C. Hb. Glob. Média.	32.3	g/dL
R.D.W.	12.3	%

LEUCOGRAMA:

Leucócitos.....	5.860	x10 ³ /uL
Neutrófilos.....	59 %	3.457
Eosinófilos.....	2 %	0.117
Basófilos.....	0 %	0.000
Linfócitos.....	31 %	1.817
Monócitos.....	8 %	0.469

Plaquetas.....	237	x10 ³ /uL
----------------	-----	----------------------

VELOCIDADE DE SEDIMENTAÇÃO

1ª hora	6	mm/h	Até 20
---------	---	------	--------

BIOQUIMICA

GLICÊMIA	85	mg/dL	75-115
CREATININA	0.8	mg/dL	até 1.4
ACIDO URICO	4.3	mg/dL	3.4 - 7.0
AST (GOT) -	25	U/L	<38
ALT (GPT) -	37	U/L	<41
GAMA GLUTAMIL TRANSPEPTIDASE	18	U/L	11 - 49



Pag.2

Exmo.Senhor

Analises requisitadas por:

Data Colheita:
Data Saída :

NT:249 NC:71745 NMec:0503100
Serviço: Fiso: Cama:

ANALISES	RESULTADOS	/ UNIDADES	VALORES REFERENCIA
COLESTEROL TOTAL	185	mg/dL	150-200 (baixo risco)
COLESTEROL - HDL	57	mg/dL	Homens < 35 (risco elevado) até 0.9 (risco elevado) Mulheres < 45 (risco elevado) até 1.15 (risco elevado)
COLESTEROL - LDL	101	mg/dL	< 130 (baixo risco)
TRIGLICERIDOS	137	mg/dL	até 150 (baixo risco)

ESTUDO DA URINA

URINA TIPO II Não veio amostra
ELEMENTOS ANORMAIS

MARCADORES DE HEPATITE B

HBs Ag Negativo
HBs Ac 75.3 UI/L Imme >=100 (E.L.I.S.A.)
HBe Ac Negativo

MARCADORES DE HEPATITE C

ANTI-HCV (met.ELISA) Negativo

SEROLOGIA DE VIH

AC.ANTI-H.I.V.I/II (met.ELISA) Negativo

MARCADORES ONCOLOGICOS

P.S.A.-AG.ESPECIFICO DA PROSTATA 0.31 ng/mL 0.0-4.0 (Mét.Quimiluminesc)



Pag.3

Exmo.Senhor

Analises requisitadas por:

Data Colheita:

Data Saida :

NT:249 NC:71745 NMec:0503100

Serviço: Piso: Cama:

ANALISES	RESULTADOS / UNIDADES	VALORES REFERENCIA
IMUNOLOGIA		
VDRL	Negativo	

